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
10/868,434	06/14/2004	Steven R. Carr	200400406-1

CONFIRMATION NO. 4452

56436
Hewlett Packard Enterprise
3404 E. Harmony Road
Mail Stop 79
Fort Collins, CO 80528

POWER OF ATTORNEY NOTICE

OC000000073543500

Date Mailed: 06/07/2024

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 06/04/2024.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/tawashington/

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
10/868,434	06/14/2004	Steven R. Carr	200400406-1

CONFIRMATION NO. 4452

192641
Botos Churchill IP Law LLP - Malik
430 Mountain Avenue
Suite 401
New Providence, NJ 07974

POA ACCEPTANCE LETTER

OC000000073543527

Date Mailed: 06/07/2024

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 06/04/2024.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/tawashington/

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
10/868,434	**06/04/2024 11:17:25 AM Z ET**	**200400406-1**

Title of Invention

DATA SERVICES HANDLER

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	7313575
CONFIRMATION #	4452	FILED BY	Gaby Hecht
PATENT CENTER #	65809921	FILING DATE	06/14/2004
CUSTOMER #	56436	FIRST NAMED INVENTOR	Steven R. Carr
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	Richard Botos

Documents

TOTAL DOCUMENTS: 4

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
7313575_statement_373c.pdf	3	Assignee showing of ownership per 37 CFR 3.73	112 KB
7313575 POA Transmittal.pdf	1	Power of Attorney	153 KB
7313575 Change of Correspondence Address Patent.pdf	2	Change of Address	265 KB
Valtrus Power of Attorney - Post AIA General.pdf	2	Power of Attorney	2368 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
7313575_statement_373c.pdf	4622128A35FAB7D5C2DF9B11886C290451859E1E7F4DAE890 D98D7FF570DFC0962CFA2C61093756C06A19469FDC0253955 AF5DCF286A27C65BC51927ED29B2BD
7313575 POA Transmittal.pdf	9B4683797E8A2B4EEDCD81A8CBCEEA17F00E47A9E7154E53 584A78106B7BE30F35324AD5CC38E8341FABACA08D07C1D9 EC5336256FC04022A9F320F85DB39090
7313575 Change of Correspondence Address Patent.pdf	8AC0B7ECEE8F2431CA975A5E99D1D4AC18965AA35AF571BA 6760697CFE93C2394D0F07BD67BC14D95C45A019A9596F78C 871309D252F194D0EDB495930C00925
Valtrus Power of Attorney - Post AIA General.pdf	193CAD07AFE8B9A537D40B3967A58851DCC29CEA84877CE1 5E6050E4B8A3F7D3F8672BD0A327006D203071F6F37D69B0FF 5708A4EDAEBDB6463AD8EB5FE7EA64

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 filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

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

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

New International Application Filed with the USPTO as a Receiving Office

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

PTO/AIA/123 (08-12)
Approved for use through 09/30/2025. OMB 0651-0035
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.

CHANGE OF CORRESPONDENCE ADDRESS
Patent

Address to:
Mail Stop Post Issue
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Patent Number	7,313,575
Issue Date	12/25/2007
Application Number	10/868,434
Filing Date	06/14/2004
First Named Inventor	Steven R. Carr
Attorney Docket Number	200400406-1

Please change the Correspondence Address for the above-identified patent to:

☑ The address associated with Customer Number: 192641

OR

☐ **Firm *or* Individual Name**

Address

City | **State** | **ZIP**

Country

Telephone | **Email**

This form cannot be used to change the data associated with a Customer Number. To change the data associated with an existing Customer Number use "Request for Customer Number Data Change" (PTO/SB/124).

This form will not affect any "fee address" provided for the above-identified patent. To change a "fee address" use the "Fee Address Indication Form" (PTO/SB/47).

I am the:

☐ Patentee.

☐ If the Patentee was not the applicant for patent (37 CFR 1.42), then a Statement under 37 CFR 3.73(c) (Form PTO/AIA/96 or equivalent) is enclosed or was filed on _____. See 37 CFR 3.71.

☑ Attorney or agent of record. Registration Number 32,016.

☐ Patent practitioner acting in a representative capacity whose correspondence address is the correspondence address of record. Notice has been given to the patentee or owner. Registration Number 32,016.

Signature /Richard J. Botos/

Typed or Printed Name Richard J. Botos

Date June 4, 2024 | Telephone (908) 738-1770

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4(d) for signature requirements and certifications. Submit multiple forms if more than one signature is required, see below*.

☑ *Total of 1 forms are submitted.

A Federal agency may not conduct or sponsor, and a person is not required to respond to, nor shall a person be subject to a penalty for failure to comply with an information collection subject to the requirements of the Paperwork Reduction Act of 1995, unless the information collection has a currently valid OMB Control Number. The OMB Control Number for this information collection is 0651-00**. Public burden for this form is estimated to average ** hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the information collection. Send comments regarding this burden estimate or any other aspect of this information collection, including suggestions for reducing this burden to the Chief Administrative Officer, United States Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450 or email InformationCollection@uspto.gov. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **If filing this completed form by mail, 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.

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. The United States Patent and Trademark Office (USPTO) collects the information in this record under authority of 35 U.S.C. 2. The USPTO's system of records is used to manage all applicant and owner information including name, citizenship, residence, post office address, and other information with respect to inventors and their legal representatives pertaining to the applicant's/owner's activities in connection with the invention for which a patent is sought or has been granted. The applicable Privacy Act System of Records Notice for the information collected in this form is COMMERCE/PAT-TM-7 Patent Application Files, available in the Federal Register at 78 FR 19243 (March 29, 2013). https://www.govinfo.gov/content/pkg/FR-2013-03-29/pdf/2013-07341.pdf

Routine uses of the information in this record may include disclosure to: 1) law enforcement, in the event that the system of records indicates a violation or potential violation of law; 2) a Federal, state, local, or international agency, in response to its request; 3) a contractor of the USPTO having need for the information in order to perform a contract; 4) the Department of Justice for determination of whether the Freedom of Information Act (FOIA) requires disclosure of the record; 5) a Member of Congress submitting a request involving an individual to whom the record pertains, when the individual has requested the Member's assistance with respect to the subject matter of the record; 6) a court, magistrate, or administrative tribunal, in the course of presenting evidence, including disclosures to opposing counsel in the course of settlement negotiations; 7) the Administrator, General Services Administration (GSA), or their designee, during an inspection of records conducted by GSA under authority of 44 U.S.C. 2904 and 2906, in accordance with the GSA regulations and any other relevant (i.e., GSA or Commerce) directive, where such disclosure shall not be used to make determinations about individuals; 8) 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)); 9) the Office of Personnel Management (OPM) for personnel research purposes; and 9) the Office of Management and Budget (OMB) for legislative coordination and clearance.

If you do not furnish the information requested on this form, the USPTO may not be able to process and/or examine your submission, which may result in termination of proceedings, abandonment of the application, and/or expiration of the patent.

Additional Uses

Additional USPTO uses of the information in this record may include disclosure to: 1) the International Bureau of the World Intellectual Property Organization, if the record is related to an international application filed under the Patent Cooperation Treaty; 2) the public i) after publication of the application pursuant to 35 U.S.C. 122(b), ii) after issuance of a patent pursuant to 35 U.S.C. 151, iii) 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 inspections, or an issued patent, or iv) without publication of the application or patent under the specific circumstances provided for by 37 CFR 1.14(a)(1)(v)-(vii); and/or 3) the National Archives and Records Administration, for inspection of records.

PTO/AIA/96 (08-12)
Approved for use through 11/30/2020. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(c)

Applicant/Patent Owner: VALTRUS INNOVATIONS LIMITED

Application No./Patent No.: 10/868,434/7,313,575 Filed/Issue Date: June 14, 2004/December 25, 2007

Titled: DATA SERVICES HANDLER

VALTRUS INNOVATIONS LIMITED, a a Corporation

(Name of Assignee) (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that, for the patent application/patent identified above, it is (choose **one** of options 1, 2, 3 or 4 below):

1. ☑ The assignee of the entire right, title, and interest.

2. ☐ An assignee of less than the entire right, title, and interest (check applicable box):

   ☐ The extent (by percentage) of its ownership interest is _____%. Additional Statement(s) by the owners holding the balance of the interest must be submitted to account for 100% of the ownership interest.

   ☐ There are unspecified percentages of ownership. The other parties, including inventors, who together own the entire right, title and interest are:

   Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

3. ☐ The assignee of an undivided interest in the entirety (a complete assignment from one of the joint inventors was made). The other parties, including inventors, who together own the entire right, title, and interest are:

   Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

4. ☐ The recipient, via a court proceeding or the like (*e.g.*, bankruptcy, probate), of an undivided interest in the entirety (a complete transfer of ownership interest was made). The certified document(s) showing the transfer is attached.

The interest identified in option 1, 2 or 3 above (not option 4) is evidenced by either (choose **one** of options A or B below):

A. ☐ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel _____, Frame _____, or for which a copy thereof is attached.

B. ☑ A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

1. From: Inventors To: HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.

   The document was recorded in the United States Patent and Trademark Office at
   Reel 015482, Frame 0250, or for which a copy thereof is attached.

2. From: HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P. To: HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP

   The document was recorded in the United States Patent and Trademark Office at
   Reel 037079, Frame 0001, or for which a copy thereof is attached.

This collection of information is required by 37 CFR 3.73(b). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450**.

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

PTO/AIA/96 (08-12)
Approved for use through 11/30/2020. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(c)

3. From: HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP and HEWLETT PACKARD ENTERPRISE COMPANY To: VALTRUS INNOVATIONS LIMITED

The document was recorded in the United States Patent and Trademark Office at
Reel 055360, Frame 0424, or for which a copy thereof is attached.

4. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached.

5. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached.

6. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached.

☐ Additional documents in the chain of title are listed on a supplemental sheet(s).

☐ As required by 37 CFR 3.73(c)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (i.e., a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

/Richard J. Botos/	June 4, 2024
Signature	Date
Richard J. Botos	32,016
Printed or Typed Name	Title or Registration Number

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.

Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82A (07-13)
Approved for use through 11/30/2014. OMB 0651-0051
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.

TRANSMITTAL FOR POWER OF ATTORNEY TO ONE OR MORE REGISTERED PRACTITIONERS

NOTE: This form is to be submitted with the Power of Attorney by Applicant form (PTO/AIA/82B) to identify the application to which the Power of Attorney is directed, in accordance with 37 CFR 1.5, unless the application number and filing date are identified in the Power of Attorney by Applicant form. If neither form PTO/AIA/82A nor form PTO/AIA82B identifies the application to which the Power of Attorney is directed, the Power of Attorney will not be recognized in the application.

Application Number	10/868,434
Filing Date	June 14, 2004
First Named Inventor	Steven R. Carr
Title	DATA SERVICES HANDLER
Art Unit	2167
Examiner Name	KUEN S LU
Attorney Docket Number	200400406-1

SIGNATURE of Applicant or Patent Practitioner

Signature	/Richard J. Botos/	Date (Optional)	June 4, 2024
Name	Richard J. Botos	Registration Number	32,016
Title (if Applicant is a juristic entity)			
Applicant Name (if Applicant is a juristic entity)			

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4(d) for signature requirements and certifications. If more than one applicant, use multiple forms.

☑ *Total of 1 forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

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

PTO/AIA/80 (07-17)
Approved for use through 01/31/2018. OMB 0651-0035
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no person is required to respond to a collection of information unless it displays a valid OMB control number

POWER OF ATTORNEY TO PROSECUTE APPLICATIONS BEFORE THE USPTO

I hereby revoke all previous powers of attorney given in the application identified in the attached statement under 37 CFR 3.73(c).

I hereby appoint:

[X] Practitioners associated with Customer Number: 192641

OR

[ ] Practitioner(s) named below (if more than ten patent practitioners are to be named, then a customer number must be used):

Name	Registration Number	Name	Registration Number

As attorney(s) or agent(s) to represent the undersigned before the United States Patent and Trademark Office (USPTO) in connection with any and all patent applications assigned only to the undersigned according to the USPTO assignment records or assignment documents attached to this form in accordance with 37 CFR 3.73(c).

Please change the correspondence address for the application identified in the attached statement under 37 CFR 3.73(c) to:

[X] The address associated with Customer Number: 192641

OR

[ ]

Firm or individual name		
Address		
City	State	Zip
Country		
Telephone	Email	

Assignee name and address: VALTRUS INNOVATIONS LIMITED THE GLASS HOUSES 92 GEORGES STREET LOWER, DUN LAOGHAIRE DUBLIN, IRELAND A96 VR66

A copy of this form, together with a statement under 37 CFR 3.73(c) (Form PTO/AIA/96 or equivalent) is required to be filed in each application in which this form is used. The statement under 37 CFR 3.73(c) may be completed by one of the practitioners appointed in this form, and must identify the application in which this Power of Attorney is to be filed.

SIGNATURE of Assignee of Record

The individual whose signature and title is supplied below is authorized to act on behalf of the assignee.

Signature /Angela Quinlan/	Date 5/22/2024
Name Angela Quinlan	Telephone +353 87 9171202
Title Managing Director	

This collection of information is required by 37 CFR 1.31, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public, which is to update (and by the USPTO to process) the file of a patent or reexamination proceeding. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 18 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.

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.

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	PATENT NUMBER	GROUP ART UNIT	FILE WRAPPER LOCATION
10/868,434	7313575	2167	7581

OC000000074018485

Correspondence Address/Fee Address Change

The following fields have been set to Customer Number 56436 on 03/16/2015

- **Correspondence Address**
- **Maintenance Fee Address**

The address of record for Customer Number 56436 is:

56436
HP Company
Intellectual Property Administration
3404 E. Harmony Road
Mail Stop 35
Fort Collins, CO 80528

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 7,313,575 B2
APPLICATION NO. : 10/868434
DATED : December 25, 2007
INVENTOR(S) : Steven R. Carr et al.

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

In column 1, lines 32–33, delete "a cross" and insert -- across --, therefor.

In column 56, line 43, in Claim 12, delete "serviet" and insert -- servlet --, therefor.

In column 56, line 49, in Claim 12, delete "serviet" and insert -- servlet --, therefor.

In column 56, line 66, in Claim 14, delete "Internal" and insert -- internal --, therefor.

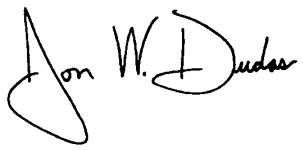
In column 59, line 20, in Claim 33, delete "a" before "at least".

In column 60, line 6, in Claim 36, delete "transfering" and insert -- transferring --, therefor.

In column 60, line 11, in Claim 36, delete "obiect" and insert -- object --, therefor.

Signed and Sealed this

Eighth Day of July, 2008



JON W. DUDAS
Director of the United States Patent and Trademark Office

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

IN RE APPLICATION OF: U.S. Patent No. 7,313,575

USPTO CONFIRMATION CODE: 4452

APPLICATION NO.: 10/868,434

FILED: June 14, 2004

EXAMINER: Kuen S. Lu

GROUP ART UNIT: 2167

FOR: DATA SERVICES HANDLER

37 CFR 1.322 & 37 CFR 1.323 REQUEST FOR CERTIFICATE OF CORRECTION

HONORABLE COMMISSIONER OF PATENTS & TRADEMARKS

Sir:

The following is a request for a certificate of correction in Serial Number 10/868,434, now Patent Number 7,313,575.

A certificate of correction under 35 USC 254 is respectfully requested in the above-identified patent.

☒ All errors were the fault of the USPTO, no fee required. In the event that a further fee is required, please charge the amount to Deposit Account No.082025.

☐ All errors were the fault of the applicant and, accordingly, please charge **$100.00** to our Deposit Account No.082025. In the event that a further fee is required, please charge the amount to the same Deposit Account.

☐ The errors were the fault of both the applicant and USPTO and, accordingly, please charge **$100.00** to our Deposit Account No. 082025. In the event that a further fee is required, please charge the amount to the same Deposit Account.

The exact locations where the errors appear in the patent and patent application are as follows:

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 15 of 302

In column 1, lines 32–33, delete "a cross" and insert - - across - -, therefor.
(Specification filed on June 14, 2004, page 1, line 20)

In column 56, line 43, in Claim 12, delete "serviet" and insert - - servlet - -, therefor.
(Amendment to claims filed on Mar. 13, 2007, page 5 of 13, in Claim 12, line 3)

In column 56, line 49, in Claim 12, delete "serviet" and insert - - servlet - -, therefor.
(Amendment to claims filed on Mar. 13, 2007, page 5 of 13, in Claim 12, line 8)

In column 56, line 66, in Claim 14, delete "Internal" and insert - - internal - -, therefor.
(Amendment to claims filed on Mar. 13, 2007, page 5 of 13, in Claim 14, line 5)

In column 59, line 20, in Claim 33, delete "a" before "at least".
(Examiner's amendment dated on June 19, 2007, page 8, in Claim 33, line 3)

In column 60, line 6, in Claim 36, delete "transfering" and insert - - transferring - -, therefor.
(Examiner's amendment dated June 19, 2007, page 8, in Claim 36, line 12)

In column 60, line 11, in Claim 36, delete "obiect" and insert - - object - -, therefor.
(Examiner's amendment dated June 19, 2007, page 9, in Claim 36, line 2)

The requested corrections are attached on Form PTO 1050.

Respectfully Submitted

5/14/08
DATE

Name: Kevin M. Hart
Registration No.: 36,823

Attorney/Agent of Record

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 16 of 302

PTO/SB/44 (04-05)
Approved for use through 04/30/2007. OMB 0651-0033
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.
(Also Form PTO-1050)

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO : 7,313,575

APPLICATION NO. : 10/868,434

ISSUE DATE : Dec. 25, 2007

INVENTOR(S) : Steven R. Carr, et al.

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

In column 1, lines 32--33, delete "a cross" and insert - - across - -, therefor.

In column 56, line 43, in Claim 12, delete "serviet" and insert - - servlet - -, therefor.

In column 56, line 49, in Claim 12, delete "serviet" and insert - - servlet - -, therefor.

In column 56, line 66, in Claim 14, delete "Internal" and insert - - internal - -, therefor.

In column 59, line 20, in Claim 33, delete "a" before "at least".

In column 60, line 6, in Claim 36, delete "transfering" and insert - - transferring - -, therefor.

In column 60, line 11, in Claim 36, delete "obiect" and insert - - object - -, therefor.

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

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

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

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

Electronic Acknowledgement Receipt

EFS ID:	3308777
Application Number:	10868434
International Application Number:	
Confirmation Number:	4452
Title of Invention:	DATA SERVICES HANDLER
First Named Inventor/Applicant Name:	Steven R. Carr
Customer Number:	22879
Filer:	Kevin M. Hart/Kathy Hungsberg
Filer Authorized By:	Kevin M. Hart
Attorney Docket Number:	200400406-1
Receipt Date:	15-MAY-2008
Filing Date:	14-JUN-2004
Time Stamp:	15:14:34
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes) /Message Digest	Multi Part /.zip	Pages (if appl.)
1	Request for Certificate of Correction	200400406.pdf	679655 034393efbe45d0dadba5c0cda38751dbc17429a3	no	3

Warnings:

The page size in the PDF is too large. The pages should be 8.5 x 11 or A4. If this PDF is submitted, the pages will be resized upon entry into the Image File Wrapper and may affect subsequent processing	
Information:	
Total Files Size (in bytes):	679655

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.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/868,434	12/25/2007	7313575	200400406-1	4452

22879 7590 12/05/2007

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

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 495 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

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

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

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

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

Steven R. Carr, San Jose, CA;
Venkatakrishnan Muthuswamy, Cupertino, CA;
Tudor Har, Santa Clara, CA;

UNITED STATES PATENT AND TRADEMARK OFFICE

OFFICE

SERIAL NUMBER	FILING DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NO.
10/868434	6/14/04	CARR, STEVEN R.	200400406-1

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONT ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS CO 80527-2400

EXAMINER	
LU, KUEN	
ART UNIT	PAPER NUMBER
2167	

DATE MAILED: 11/16/07

NOTICE OF RESCINDED ABANDONMENT

☒ **In response to your communication filed** 11/5/07

☐ **Through inadvertence, a Notice of Abandonment was mailed in the above-identified application. The Notice of Abandonment is hereby rescinded.**

for Kenneth Sprunk

Manager, Publications Division
Office of Patent Publication
571-272-4200

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 21 of 302

RECEIVED
CENTRAL FAX CENTER
NOV 05 2007

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200400406-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Steven Robert Carr et al

Confirmation No.: 4452

Application No.: 10/868434

Examiner:

Filing Date: Jun 14, 2004

Group Art Unit:

Title: Data Services Handler

Mail Stop Petition
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

PETITION FOR WITHDRAWAL OF ABANDONMENT

1. Applicant petitions that the abandonment set forth in the notice mailed by the office on 09/13/07 be withdrawn.

2. Submitted herewith is:

- [x] A copy of the page of the response mailed on 06/28/07 showing a Certificate of Mailing executed on 06/28/07.
- [ ] A copy of the post card identifying the papers filed and showing the US PTO receipt stamp dated _____.
- [ ] A copy of the complete response previously filed.
- [ ] A copy of the attorney's Deposit Account Statement in which the item corresponding to the response referred to above is checked.
- [ ] A statement as setforth in 37 CFR 1.8(b)(3).

3. Please proceed with further examination of this application on the basis of:

- [ ] The original papers filed, which have now reached the appropriate area of the PTO, and/or
- [x] The attached copy of the papers originally filed.

At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees.

- [ ] A duplicate copy of this transmittal letter is enclosed.

- [ ] I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450
  Date of Deposit:

OR

- [x] I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number (571)273-8300.
  Date of facsimile: 11/05/07
  Typed Name: Kathy L. Hungsberg
  Signature: Kathy L. Hungsberg

Respectfully submitted,
Steven Robert Carr et al
By Kevin Hart

Kevin Hart
Attorney/Agent for Applicant(s)
Reg No. : 36,623
Date : 11/05/07
Telephone : 970 898 7057

Rev 10/07 (AbdWhd)

Auto-Reply Facsimile Transmission

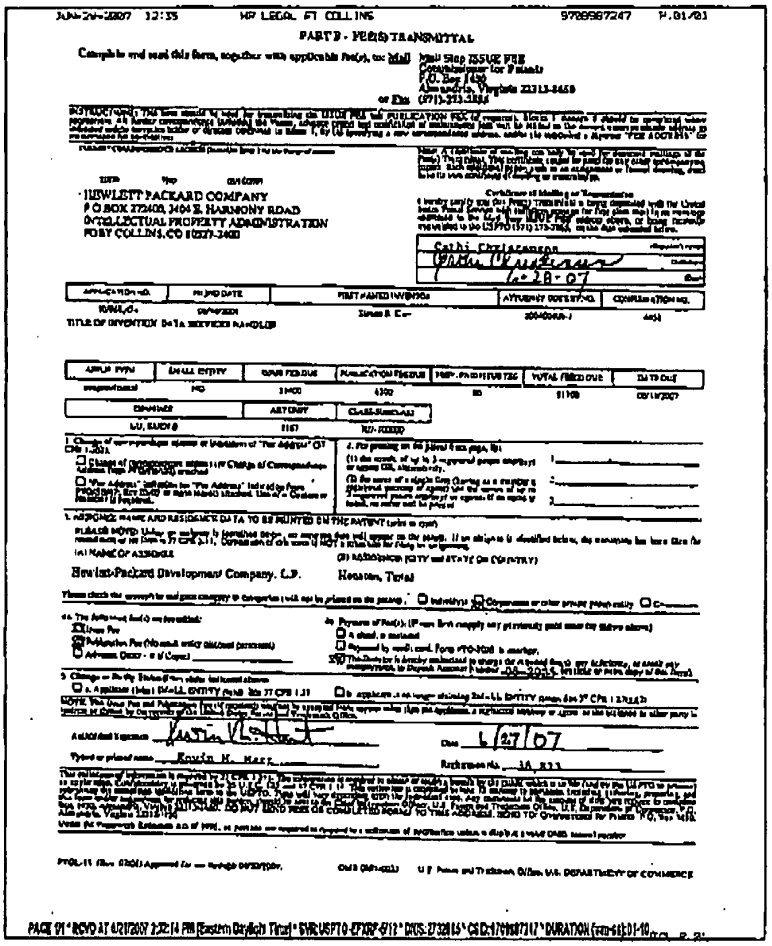
RECEIVED
CENTRAL FAX CENTER
NOV 0 5 2007

TO: Fax Sender at 9708987247

Fax Information
Date Received: 6/28/2007 2:32:14 PM [Eastern Daylight Time]
Total Pages: 1 (including cover page)

ADVISORY: This is an automatically generated return receipt confirmation of the facsimile transmission received by the Office. Please check to make sure that the number of pages listed as received in Total Pages above matches what was intended to be sent. Applicants are advised to retain this receipt in the unlikely event that proof of this facsimile transmission is necessary. Applicants are also advised to use the certificate of facsimile transmission procedures set forth in 37 CFR 1.8(a) and (b), 37 CFR 1.6(f). Trademark Applicants, also see the Trademark Manual of Examining Procedure (TMEP) section 306 et seq.

Received
Cover
Page
=======>



PART B - FEE(S) TRANSMITTAL

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

Cathi Christensen
6-28-07

Hewlett-Packard Development Company, L.P. Houston, Texas

Kevin K. Harr 6/27/07

PAGE 1/3 * RCVD AT 6/28/2007 2:32:14 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/4 * DNIS:2738300 * CSID:9708987247 * DURATION (mm-ss):01-10

MODE = MEMORY TRANSMISSION START=JUN-28 12:35 END=JUN-28 12:36

FILE NO.=869

RECEIVED
CENTRAL FAX CENTER
NOV 0 5 2007

STN NO.	COMM.	ONE-TOUCH/ ABBR NO.	STATION NAME/TEL NO.	PAGES	DURATION
001	OK	<25>	PTO ISSUE FEES	001/001	00:00:41

-HP LEGAL FT COLLINS -

******************************* -9708987247 - ***** - 9708987247- *********

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)

22879 7590 05/16/2007

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

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.

Cathi Christensen (Depositor's name)
[signature] (Signature)
6-28-07 (Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/868,434	06/14/2004	Steven R. Carr	200400406-1	4452

TITLE OF INVENTION: DATA SERVICES HANDLER

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$300	$0	$1700	08/16/2007

EXAMINER	ART UNIT	CLASS-SUBCLASS
LU, KUEN S	2167	707-100000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. Use of a Customer Number is required.

2. For printing on the patent front page, list
(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
(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)
Hewlett-Packard Development Company, L.P. Houston, Texas

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 08-2025 (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 [signature] Date 6/27/07

Typed or printed name Kevin M. Hart Registration No. 36,823

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.

PTOL-85 (Rev. 07/06) Approved for use through 04/30/2007. OMB 0651-0033 U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

PAGE 3/3 * RCVD AT 11/5/2007 3:21:21 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-1/4 * DNIS:2738300 * CSID:9708987247 * DURATION (mm-ss):01-12

UNITED STATES PATENT AND TRADEMARK OFFICE

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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/868,434	06/14/2004	Steven R. Carr	200400406-1	4452

22879 7590 09/13/2007

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
LU, KUEN S	
ART UNIT	**PAPER NUMBER**
2167	
MAIL DATE	**DELIVERY MODE**
09/13/2007	PAPER

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

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

Notice of Abandonment	Application No.	Applicant(s)	
	10/868,434	CARR	
	Examiner	Art Unit	
	LU	2167	

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

This application is abandoned in view of:

1. ☐ Applicant's failure to timely file a proper reply to the Office letter mailed on _____.
   (a) ☐ A reply was received on _____ (with a Certificate of Mailing or Transmission dated _____), which is after the expiration of the period for reply (including a total extension of time of _____ month(s)) which expired on _____.
   (b) ☐ A proposed reply was received on _____, but it does not constitute a proper reply under 37 CFR 1.113 (a) to the final rejection.
   (A proper reply under 37 CFR 1.113 to a final rejection consists only of: (1) a timely filed amendment which places the application in condition for allowance; (2) a timely filed Notice of Appeal (with appeal fee); or (3) a timely filed Request for Continued Examination (RCE) in compliance with 37 CFR 1.114).
   (c) ☐ A reply was received on _____ but it does not constitute a proper reply, or a bona fide attempt at a proper reply, to the non-final rejection. See 37 CFR 1.85(a) and 1.111. (See explanation in box 7 below).
   (d) ☐ No reply has been received.

2. ☒ Applicant's failure to timely pay the required issue fee and publication fee, if applicable, within the statutory period of three months from the mailing date of the Notice of Allowance (PTOL-85).
   (a) ☐ The issue fee and publication fee, if applicable, was received on _____ (with a Certificate of Mailing or Transmission dated _____), which is after the expiration of the statutory period for payment of the issue fee (and publication fee) set in the Notice of Allowance (PTOL-85).
   (b) ☐ The submitted fee of $_____ is insufficient. A balance of $_____ is due.
   The issue fee required by 37 CFR 1.18 is $_____. The publication fee, if required by 37 CFR 1.18(d), is $_____.
   (c) ☒ The issue fee and publication fee, if applicable, has not been received.

3. ☐ Applicant's failure to timely file corrected drawings as required by, and within the three-month period set in, the Notice of Allowability (PTO-37).
   (a) ☐ Proposed corrected drawings were received on _____ (with a Certificate of Mailing or Transmission dated _____), which is after the expiration of the period for reply.
   (b) ☐ No corrected drawings have been received.

4. ☐ The letter of express abandonment which is signed by the attorney or agent of record, the assignee of the entire interest, or all of the applicants.

5. ☐ The letter of express abandonment which is signed by an attorney or agent (acting in a representative capacity under 37 CFR 1.34(a)) upon the filing of a continuing application.

6. ☐ The decision by the Board of Patent Appeals and Interference rendered on _____ and because the period for seeking court review of the decision has expired and there are no allowed claims.

7. ☐ The reason(s) below:

/lgd/

Petitions to revive under 37 CFR 1.137(a) or (b), or requests to withdraw the holding of abandonment under 37 CFR 1.181, should be promptly filed to minimize any negative effects on patent term.

01 FC:1051 130.00 DA
11/1/2007 HPHE 00000001 10504504
520280 08/29/2005 [illegible]

MODE = MEMORY TRANSMISSION START-JUN-28 12:35 END-JUN-29 12:36

FILE NO.-869

STN NO.	COMM.	ONE-TOUCH/ ABBR NO.	STATION NAME/TEL NO.	PAGES	DURATION
001	OK	(25)	PTO ISSUE FEES	001/001	00:00'41

RECEIVED
CENTRAL FAX CENTER
NOV 05 2007

-HP LEGAL FT COLLINS -

-9708987247 - - 9708987247-

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)

22879 7590 05/16/2007

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

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.

Cathi Christensen (Depositor's name)
(Signature)
6-28-07 (Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/858,434	06/14/2004	Steven R. Carr	200400408-1	4453

TITLE OF INVENTION: DATA SERVICES HANDLER

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$300	$0	$1700	09/16/2007

EXAMINER	ART UNIT	CLASS-SUBCLASS
LU, KUEN S	2157	707-100000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. Use of a Customer Number is required.

2. For printing on the patent front page, list
(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
(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: Hewlett-Packard Development Company, L.P.
(B) RESIDENCE: (CITY and STATE OR COUNTRY): Houston, Texas

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 08-2025 (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 6/27/07

Typed or printed name Kevin M. Hart Registration No. 36,823

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.

PTOL-85 (Rev. 07/05) Approved for use through 04/30/2007. OMB 0651-0033 U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

5

UNITED STATES PATENT AND TRADEMARK OFFICE

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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/868,434	06/14/2004	Steven R. Carr	200400406-1	4452

22879 7590 06/19/2007
HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
LU, KUEN S	
ART UNIT	**PAPER NUMBER**
2167	
MAIL DATE	**DELIVERY MODE**
06/19/2007	PAPER

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

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

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 28 of 302

Supplemental Notice of Allowability	Application No. 10/868,434	Applicant(s) CARR ET AL.	
	Examiner Kuen S. Lu	Art Unit 2167	

-- 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 *3/13/2007*.

2. ☒ The allowed claim(s) is/are *1-36 and 40 (renumbered to 1-37)*.

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 *5/11/2007* .
7. ☒ Examiner's Amendment/Comment
8. ☐ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 29 of 302

DETAILED ACTION

1. This action is responsive to Applicant's Amendment filed March 13, 2007. It is acknowledged that claims 1, 4, 9, 22, 28, 33 and 36 were amended and 37, 38 and 39 were canceled.

As necessitated by the Amendment, Examiner herby withdraws rejections of Claims 22-27 and 38-39 under 35 U.S.C. §112, first paragraph; rejection of Claim 36 under 35 U.S.C. §112, first paragraph; rejections of Claims 1-39 under 35 U.S.C. §101 and objections to claims 9 and 37.

2. After a thorough search and examination of the present application, and in light of the prior art made of record, Applicant's Amendment and Remarks filed March 13, 2007and Examiner's amendments made May 11, 2007, Claims 1-36 and 40 (renumbered to 1-37) are allowed.

Examiner's Amendments

3. Authorization for this Examiner's amendment, listed below, was given in telephone interview with Applicant, Mr. Ken J. Koestner (Registration 33,004) on May 11, 2007 for amending claims 1, 7, 10 and 15. The summary of interview conducted on May 4, 2007 is attached.

3.1. Please amend Claims 1, 2, 4, 22-28, 33, 36 and 40 as follow:

1. (Currently Amended)): A data services handler <u>for execution on a computing system</u> comprising:

an interface executing on the computing system for communicating between

a data store and applications that supply and consume data; and

a real time information director (RTID) ~~executable~~ executing on the computing system that transforms data for supply, ~~and/or~~ consumption, or both by the applications under direction of polymorphic metadata that defines a security model and data integrity rules for application to the data, the RTID ~~configured~~ to dynamically instantiate, create, and cache for reuse disassemblers and assemblers according to demand, the RTID further comprising:

at least one disassembler defined according to the polymorphic metadata that disassembles inbound data documents into multiple records in preparation for insertion into metadata tables; and

at least one assembler defined according to the polymorphic metadata that assembles outbound data documents from at least one record selected from at least one metadata-selected table whereby data is transformed for supply, consumption or both.

2. (Currently amended): The data services handler according to Claim 1 further comprising:

a data mapper ~~capable of~~ collecting data from a plurality of diverse applications that can supply data in multiple diverse external document formats and mapping the supplied data to a predetermined unified data model according to the polymorphic metadata.

4. (Currently amended): The data services handler according to Claim 1 wherein the real time information director further comprises:

the RTID ~~configured to use~~ using a same descriptive polymorphic metadata to build both assemblers and disassemblers for a same document class.

22. (Currently Amended): An article of manufacture comprising:

a controller-usable storage medium having a ~~computable~~ computer readable program code embodied therein for managing data services, the ~~computable~~ computer readable program code further comprising:

a code ~~capable of~~ causing the controller to transfer data among a data store and applications that supply and consume data;

a code ~~capable of~~ causing the controller to transform data for transfer using a representation of metadata as polymorphic objects, enabling individual metadata objects to have distinct numbers and types of attributes according to a predetermined object class, the behaviors and attributes being defined specific to object class;

a code ~~capable of~~ causing the controller to dynamically instantiate, create, and cache for reuse at least one disassembler and at least one assembler according to demand~~,~~;

a code dynamically created by the controller as defined by polymorphic metadata that disassembles inbound data documents into multiple records in preparation for insertion into metadata tables; and

a code dynamically created by the controller as defined by polymorphic metadata that assembles outbound data documents from at least one record selected from at least one metadata-selected table.

23. (Currently amended): The article of manufacture according to Claim 22 further comprising:

a code ~~capable of~~ causing the controller to transform data under direction of polymorphic metadata that defines a security model and data integrity rules for application to the data.

24. (Currently amended): The article of manufacture according to Claim 22 further comprising: a code ~~capable of~~ causing the controller to collect data from a plurality of diverse applications that can supply data in multiple diverse external document formats and mapping the supplied data to a predetermined unified data model according to the polymorphic metadata.

25. (Currently amended): The article of manufacture according to Claim 22 further comprising:

a code ~~capable of~~ causing the controller to represent metadata as stateless polymorphic objects that are permanent after initial construction, whereby methods of stateful metadata classes delegate operation to the stateless polymorphic metadata objects.

26. (Currently amended): The article of manufacture according to Claim 22 further comprising:

a code ~~capable of~~ causing the controller to construct at least one metadata object using constructors of multiple metadata classes;

a code ~~capable of~~ causing the controller to use constructor parameters to specify metadata data elements and being nested to represent hierarchical relationships between the specified metadata data elements, the at least one metadata object including identifying information for deriving the hierarchical relationships.

27. (Currently amended): The article of manufacture according to Claim 22 further comprising:

a code ~~capable of~~ causing the controller to use received data document mapping metadata to disassemble elements of inbound data documents into rows and columns to be inserted into inbound metadata-selected tables in a database, select rows and columns from outbound metadata-selected tables of the database and assemble elements from the outbound metadata-selected tables into an outbound data document, and generate a schema for documents that can be input or output using the received data document mapping metadata.

28. (Currently Amended): An enterprise system comprising:

a plurality of interlinked computer servers and networks enabling multiple users to

access applications and data on the servers; and

a real time information director (RTID) executing on at least one of the interlinked servers that manages information flow between a data store and applications that supply and consume data and dynamically instantiates, creates, and caches for reuse disassemblers and assemblers according to demand,

the RTID ~~configured to create~~ that creates at least one disassembler defined according to the polymorphic metadata that disassembles inbound data documents into multiple records in preparation for insertion into metadata tables and create at least one assembler defined according to the polymorphic metadata that assembles outbound data documents from at least one record selected from at least one metadata-selected table whereby data is transformed for supply, consumption, or both.

33. (Currently Amended): A data services handler according to Claim 1 further comprising:

~~a real time information director (RTID) executable on the computing system that maps documents into a data model further comprising:~~

~~a disassembly logic that is dynamically created by the RTID as defined by polymorphic metadata and disassembles documents for loading into a database;~~

~~an assembly logic that is dynamically created by the RTID as defined by polymorphic metadata and assembles documents from~~

~~records in the database whereby data is transformed for supply and/or consumption; and~~

~~a~~ at least one generator ~~logic~~ that is dynamically created by the RTID as defined by polymorphic metadata and generates an eXtended Markup Language (XML) Schema that describes format of the documents, the ~~disassembly~~ at least one disassembler, ~~assembly~~ assembler, and generator ~~logic being operative using~~ that use a single descriptive polymorphic metadata.

36. (Currently Amended): A method for ~~generating native documents as an extended Markup Language (XML) representation of data in a Structured Query Language (SQL) database, the method~~ executing on a computer system for managing data services comprising:

~~dynamically creating metadata for a new native document as defined by polymorphic metadata further comprising: determining whether inbound documents are inserted, delserted or upserted;~~

~~generating native documents;~~

~~determining what data from the SQL database is sufficient for usage in the new native document; and~~

~~creating an XML Schema for the native document~~

transfering data among a data store and applications that supply and consume data;

transforming data for transfer using a representation of metadata as polymorphic

objects, enabling individual metadata objects to have distinct numbers and types of attributes according to a predetermined object class, the behaviors and attributes being defined specific to object class;

dynamically instantiating, creating, and caching for reuse at least one disassembler and at least one assembler according to demand;

disassembling inbound data documents into multiple records in preparation for insertion into metadata tables; and

assembling outbound data documents from at least one record selected from at least one metadata-selected table.

40. (New): A method according to Claim 36 further comprising:

dynamically creating metadata for a new native document as defined by polymorphic metadata further comprising: determining whether inbound documents are inserted, delserted or upserted;

generating native documents;

determining what data from the SQL database is sufficient for usage in the new native document; and

creating an XML Schema for the native document.

Reasons For Allowance

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

In the Examiner's Office Action for Final Rejection of December 13, 2006, Examiner's

Claim rejection under 35 U.S.C. §103(a) is primarily based on Brendle et al: "APPLICATION BUSINESS OBJECT PROCESSING", U.S. Patent Application 2005/0021354, published January 27, 2005, hereafter "Brendle"; in view of Reed et al: "COMPUTER-BASED COMMUNICATION SYSTEM AND METHOD USING METADATA DEFINING A CONTROL-STRUCTURE", U.S. Patent 6,088,717, issued July 11, 2000, hereafter "Reed". In addition, Examiner cited Oracle® product for supporting teachings missing from the two references in order to reject every element of all claims.

After further consideration of Applicant's and Examiner's amendments made to the claims, Examiner is persuaded that the mostly recently amended claims 1, 22, 28 and 36 where *data transformation under the direction of polymorphic metadata defining a security model and integrity security rules for application data and a data structure further instantiating, creating and caching for reuse disassemblers and assemblers according to demand and in which data assembled and disassembled defined by polymorphic metadata,* a feature is further described and highlighted below, is distinct from prior art cited.

transforming data for transfer using a representation of metadata as polymorphic objects, enabling individual metadata objects to have distinct numbers and types of attributes according to a predetermined object class, the behaviors and attributes being defined specific to object class;

dynamically instantiating, creating, and caching for reuse at least one disassembler and at least one assembler according to demand; disassembling inbound data documents into multiple records in preparation for insertion into metadata tables

An updated search for the prior arts on EAST database and on domains (NPL-ACM, Google, NPL-IEEE) has been conducted. The prior arts searched and investigated in the database and domains does not fairly teach or suggest the teaching of the newly amended claimed subject matter as combined and described in each of the independent claims 1, 22, 28 and 36.

The dependent claim(s) in the groups (2-21), (23-27), (29-35) and (40), depending directly or indirectly upon claims 1, 22, 28 and 36, respectively, are also distinct from the prior art for the same reason.

5. 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."

Contact Information

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kuen S. Lu whose telephone number is (571) 272-

4114. The examiner can normally be reached on Monday-Friday (8:00 am-5:00 pm). If attempts to reach the examiner by telephone pre unsuccessful, the examiner's Supervisor, John Cottingham can be reached on (571) 272-7079. The fax phone number for the organization where this application or proceeding is assigned is 703-305-3900.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for Page 13 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 703-305-3900 (toll free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, please call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Kuen S. Lu,

Patent Examiner, Art Unit 2167

June 16, 2007

Interview Summary	Application No.	Applicant(s)	
	10/868,434	CARR ET AL.	
	Examiner	Art Unit	
	Kuen S. Lu	2167	

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

(1) *Kuen S. Lu*. (3)_____.

(2) *Mr. Ken J. Koestner (Registration 33,004)*. (4)_____.

Date of Interview: *11 May 2007*.

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

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

Claim(s) discussed: *1-40*.

Identification of prior art discussed: *n/a*.

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

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: *Applicant agreed to an Examiner's Amendment to amend claims 1, 2, 4, 22-28, 33, 36 and 40. Examiner agreed to conduct an update search on prior art and submit, pending update search result, the application for allowance review and approval.*.

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

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

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



Examiner's signature, if required

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.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 42 of 302

Notice of References Cited	Application/Control No. 10/868,434	Applicant(s)/Patent Under Reexamination CARR ET AL.	
	Examiner Kuen S. Lu	Art Unit 2167	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,088,717	07-2000	Reed et al.	709/201
*	B	US-2005/0021354	01-2005	Brendle et al.	705/001
*	C	US-2005/0091276	04-2005	Brunswig et al.	707/104.1
*	D	US-7,062,500	06-2006	Hall et al.	707/102
*	E	US-2005/0021998	01-2005	Fiedler et al.	713/200
*	F	US-2005/0044165	02-2005	O'Farrell et al.	709/213
*	G	US-2004/0045014	03-2004	Radhakrishnan, Rakesh	719/328
	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.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 43 of 302

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10868434	CARR ET AL.
	Examiner	Art Unit
	Lu, Kuen S	2167

ORIGINAL		INTERNATIONAL CLASSIFICATION	
CLASS	SUBCLASS	CLAIMED	NON-CLAIMED
707	104.1	G 0 6 F 17/00	
		G 0 6 F 7/00	
CROSS REFERENCE(S)		G 0 6 F 15/16	

CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)				
707	101	102			
709	201				

Lu, Kuen S	5/12/2007		Total Claims Allowed:	
(Assistant Examiner)	(Date)		37	
(Legal Instruments Examiner)	(Date)	6/16/2007 (Primary Examiner) (Date)	O.G. Print Claim(s) 1	O.G. Print Figure 3A

U.S. Patent and Trademark Office

Part of Paper No. 10868434-051107

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10868434	CARR ET AL.
	Examiner	Art Unit
	Lu, Kuen S	2167

✓	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	05/12/2007								
1	1	=								
2	2	=								
3	3	=								
4	4	=								
5	5	=								
6	6	=								
7	7	=								
8	8	=								
9	9	=								
10	10	=								
11	11	=								
12	12	=								
13	13	=								
14	14	=								
15	15	=								
16	16	=								
17	17	=								
18	18	=								
19	19	=								
20	20	=								
21	21	=								
22	22	=								
23	23	=								
24	24	=								
25	25	=								
26	26	=								
27	27	=								
28	28	=								
29	29	=								
30	30	=								
31	31	=								
32	32	=								
33	33	=								
34	34	=								
35	35	=								
36	36	=								

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10868434	CARR ET AL.
	Examiner	Art Unit
	Lu, Kuen S	2167

✓	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	05/12/2007								
	37	_								
	38	_								
	39	_								
37	40	=								

Search Notes

Application/Control No.	Applicant(s)/Patent under Reexamination	
10/868,434	CARR ET AL.	
Examiner	**Art Unit**	
Kuen S. Lu	2167	

SEARCHED			
Class	Subclass	Date	Examiner
707	104.1	5/12/2007	KLU
705	1	5/12/2007	KLU
709	201	5/12/2007	KLU
707	100	5/12/2007	KLU
707	102	5/12/2007	KLU
707	101	5/12/2007	KLU

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner
707	**104.1**	**5/12/2007**	**KLU**
707	102, 101	5/12/2007	KLU
707	100, 201	5/12/2007	KLU
709/201 705/1		5/12/2007	KLU

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	DATE	EXMR
consult with Hung Pham, Examiner of Art Unit 2168, concerning potential 35 USC 101 issues	11/25/2006	KLU
EAST(11/25-28/2006)	5/12/2007	KLU
NPL (google.com, IEEE, ACM)	5/12/2007	KLU
case reviewed by Mr. John Cottingham, Supervisory Primary Examiner, Art Unit 2167	5/14/2007	KLU

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 47 of 302

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

22879 7590 05/16/2007

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
LU, KUEN S	
ART UNIT	PAPER NUMBER
2167	

DATE MAILED: 05/16/2007

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/868,434	06/14/2004	Steven R. Carr	200400406-1	4452

TITLE OF INVENTION: DATA SERVICES HANDLER

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$300	$0	$1700	08/16/2007

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.

PTOL-85 (Rev. 07/06) Approved for use through 04/30/2007.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 48 of 302

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)

22879 7590 05/16/2007

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

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

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

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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/868,434	06/14/2004	Steven R. Carr	200400406-1	4452

TITLE OF INVENTION: DATA SERVICES HANDLER

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$300	$0	$1700	08/16/2007

EXAMINER	ART UNIT	CLASS-SUBCLASS
LU, KUEN S	2167	707-100000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
(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.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 49 of 302

UNITED STATES PATENT AND TRADEMARK OFFICE

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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/868,434	06/14/2004	Steven R. Carr	200400406-1	4452

22879 7590 05/16/2007

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
LU, KUEN S	
ART UNIT	PAPER NUMBER
2167	

DATE MAILED: 05/16/2007

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 486 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 486 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.

PTOL-85 (Rev. 07/06) Approved for use through 04/30/2007.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 50 of 302

Notice of Allowability	Application No.	Applicant(s)	
	10/868,434	CARR ET AL.	
	Examiner	Art Unit	
	Kuen S. Lu	2167	

-- 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 *3/13/2007*.

2. ☒ The allowed claim(s) is/are *1-36 and 40 (renumbered to 1-37)*.

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 *5/11/2007* .
7. ☒ Examiner's Amendment/Comment
8. ☐ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 51 of 302

DETAILED ACTION

1. This action is responsive to Applicant's Amendment filed March 13, 2007. It is acknowledged that claims 1, 4, 9, 22, 28, 33 and 36 were amended and 37, 38 and 39 were canceled.

As necessitated by the Amendment, Examiner herby withdraws rejections of Claims 22-27 and 38-39 under 35 U.S.C. §112, first paragraph; rejection of Claim 36 under 35 U.S.C. §112, first paragraph; rejections of Claims 1-39 under 35 U.S.C. §101 and objections to claims 9 and 37.

2. After a thorough search and examination of the present application, and in light of the prior art made of record, Applicant's Amendment and Remarks filed March 13, 2007and Examiner's amendments made May 11, 2007, Claims 1-36 and 40 (renumbered to 1-37) are allowed.

Examiner's Amendments

3. Authorization for this Examiner's amendment, listed below, was given in telephone interview with Applicant, Mr. Ken J. Koestner (Registration 33,004) on May 11, 2007 for amending claims 1, 7, 10 and 15. The summary of interview conducted on May 4, 2007 is attached.

3.1. Please amend Claims 1, 2, 4, 22-28, 33, 36 and 40 as follow:

1. (Currently Amended)): A data services handler <u>for execution on a computing system</u> comprising:

an interface executing on the computing system for communicating between

a data store and applications that supply and consume data; and

a real time information director (RTID) ~~executable~~ executing on the computing system that transforms data for supply, ~~and/or~~ consumption, or both by the applications under direction of polymorphic metadata that defines a security model and data integrity rules for application to the data, the RTID ~~configured~~ to dynamically instantiate, create, and cache for reuse disassemblers and assemblers according to demand, the RTID further comprising:

at least one disassembler defined according to the polymorphic metadata that disassembles inbound data documents into multiple records in preparation for insertion into metadata tables; and

at least one assembler defined according to the polymorphic metadata that assembles outbound data documents from at least one record selected from at least one metadata-selected table whereby data is transformed for supply, consumption or both.

2. (Currently amended): The data services handler according to Claim 1 further comprising:

a data mapper ~~capable of~~ collecting data from a plurality of diverse applications that can supply data in multiple diverse external document formats and mapping the supplied data to a predetermined unified data model according to the polymorphic metadata.

4. (Currently amended): The data services handler according to Claim 1 wherein the real time information director further comprises:

the RTID ~~configured to use~~ using a same descriptive polymorphic metadata to build both assemblers and disassemblers for a same document class.

22. (Currently Amended): An article of manufacture comprising:

a controller-usable storage medium having a ~~computable~~ computer readable program code embodied therein for managing data services, the ~~computable~~ computer readable program code further comprising:

a code ~~capable of~~ causing the controller to transfer data among a data store and applications that supply and consume data;

a code ~~capable of~~ causing the controller to transform data for transfer using a representation of metadata as polymorphic objects, enabling individual metadata objects to have distinct numbers and types of attributes according to a predetermined object class, the behaviors and attributes being defined specific to object class;

a code ~~capable of~~ causing the controller to dynamically instantiate, create, and cache for reuse at least one disassembler and at least one assembler according to demand;

a code dynamically created by the controller as defined by polymorphic metadata that disassembles inbound data documents into multiple records in preparation for insertion into metadata tables; and

a code dynamically created by the controller as defined by polymorphic metadata that assembles outbound data documents from at least one record selected from at least one metadata-selected table.

23. (Currently amended): The article of manufacture according to Claim 22 further comprising:

a code ~~capable of~~ causing the controller to transform data under direction of polymorphic metadata that defines a security model and data integrity rules for application to the data.

24. (Currently amended): The article of manufacture according to Claim 22 further comprising: a code ~~capable of~~ causing the controller to collect data from a plurality of diverse applications that can supply data in multiple diverse external document formats and mapping the supplied data to a predetermined unified data model according to the polymorphic metadata.

25. (Currently amended); The article of manufacture according to Claim 22 further comprising:

a code ~~capable of~~ causing the controller to represent metadata as stateless polymorphic objects that are permanent after initial construction, whereby methods of stateful metadata classes delegate operation to the stateless polymorphic metadata objects.

26. (Currently amended): The article of manufacture according to Claim 22 further comprising:

a code ~~capable of~~ causing the controller to construct at least one metadata object using constructors of multiple metadata classes;

a code ~~capable of~~ causing the controller to use constructor parameters to specify metadata data elements and being nested to represent hierarchical relationships between the specified metadata data elements, the at least one metadata object including identifying information for deriving the hierarchical relationships.

27. (Currently amended); The article of manufacture according to Claim 22 further comprising:

a code ~~capable of~~ causing the controller to use received data document mapping metadata to disassemble elements of inbound data documents into rows and columns to be inserted into inbound metadata-selected tables in a database, select rows and columns from outbound metadata-selected tables of the database and assemble elements from the outbound metadata-selected tables into an outbound

28. (Currently Amended): An enterprise system comprising:

a plurality of interlinked computer servers and networks enabling multiple users to access applications and data on the servers; and

a real time information director (RTID) executing on at least one of the interlinked

servers that manages information flow between a data store and applications that supply and consume data and dynamically instantiates, creates, and caches for reuse disassemblers and assemblers according to demand,

the RTID ~~configured to create~~ that creates at least one disassembler defined according to the polymorphic metadata that disassembles inbound data documents into multiple records in preparation for insertion into metadata tables and create at least one assembler defined according to the polymorphic metadata that assembles outbound data documents from at least one record selected from at least one metadata-selected table whereby data is transformed for supply, consumption, or both.

33. (Currently Amended): A data services handler according to Claim 1 further comprising:

~~a real-time information director (RTID) executable on the computing system that maps documents into a data model further comprising:~~

~~a disassembly logic that is dynamically created by the RTID as defined by polymorphic metadata and disassembles documents for loading into a database;~~

~~an assembly logic that is dynamically created by the RTID as defined by polymorphic metadata and assembles documents from records in the database whereby data is transformed for supply and/or consumption; and~~

~~a~~ at least one generator ~~logic~~ that is dynamically created by the RTID as defined by polymorphic metadata and generates an eXtended Markup Language (XML) Schema that describes format of the documents, the ~~disassembly~~ at least one disassembler, ~~assembly~~ assembler, and generator ~~logic being operative using~~ that use a single descriptive polymorphic metadata.

36. (Currently Amended): A method for ~~generating native documents as an extended Markup Language (XML) representation of data in a Structured Query Language (SQL) database, the method~~ executing on a computer system for managing data services comprising:

~~dynamically creating metadata for a new native document as defined by polymorphic metadata further comprising: determining whether inbound documents are inserted, delserted or upserted;~~

~~generating native documents;~~

~~determining what data from the SQL database is sufficient for usage in the new native document; and~~

~~creating an XML Schema for the native document~~

transfering data among a data store and applications that supply and consume data;

transforming data for transfer using a representation of metadata as polymorphic objects, enabling individual metadata objects to have distinct numbers and types of attributes according to a predetermined object class, the behaviors

and attributes being defined specific to object class;

dynamically instantiating, creating, and caching for reuse at least one disassembler and at least one assembler according to demand;

disassembling inbound data documents into multiple records in preparation for insertion into metadata tables; and

assembling outbound data documents from at least one record selected from at least one metadata-selected table.

40. (New): A method according to Claim 36 further comprising:

dynamically creating metadata for a new native document as defined by polymorphic metadata further comprising: determining whether inbound documents are inserted, delserted or upserted;

generating native documents;

determining what data from the SQL database is sufficient for usage in the new native document; and

creating an XML Schema for the native document.

Reasons For Allowance

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

In the Examiner's Office Action for Final Rejection of December 13, 2006, Examiner's Claim rejection under 35 U.S.C. §103(a) is primarily based on Brendle et al: "APPLICATION BUSINESS OBJECT PROCESSING", U.S. Patent Application 2005/0021354, published January 27, 2005, hereafter "Brendle"; in view of Reed et al:

"COMPUTER-BASED COMMUNICATION SYSTEM AND METHOD USING METADATA DEFINING A CONTROL-STRUCTURE", U.S. Patent 6,088,717, issued July 11, 2000, hereafter "Reed". In addition, Examiner cited Oracle® product for supporting teachings missing from the two references in order to reject every element of all claims.

After further consideration of Applicant's and Examiner's amendments made to the claims, Examiner is persuaded that the mostly recently amended claims 1, 22, 28 and 36 where *data transformation under the direction of polymorphic metadata defining a security model and integrity security rules for application data and a data structure further instantiating, creating and caching for reuse disassemblers and assemblers according to demand and in which data assembled and disassembled defined by polymorphic metadata,* a feature is further described and highlighted below, is distinct from prior art cited.

transforming data for transfer using a representation of metadata as polymorphic objects, enabling individual metadata objects to have distinct numbers and types of attributes according to a predetermined object class, the behaviors and attributes being defined specific to object class;

dynamically instantiating, creating, and caching for reuse at least one disassembler and at least one assembler according to demand;

disassembling inbound data documents into multiple records in preparation for

insertion into metadata tables

An updated search for the prior arts on EAST database and on domains (NPL-ACM, Google, NPL-IEEE) has been conducted. The prior arts searched and investigated in the database and domains does not fairly teach or suggest the teaching of the newly amended claimed subject matter as combined and described in each of the independent claims 1, 22, 28 and 36.

The dependent claim(s) in the groups (2-21), (23-27), (29-35) and (40), depending directly or indirectly upon claims 1, 22, 28 and 36, respectively, are also distinct from the prior art for the same reason.

5. 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."

Contact Information

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kuen S. Lu whose telephone number is (571) 272-4114. The examiner can normally be reached on Monday-Friday (8:00 am-5:00 pm). If attempts to reach the examiner by telephone pre unsuccessful, the examiner's Supervisor, John Cottingham can be reached on (571) 272-7079. The fax phone

number for the organization where this application or proceeding is assigned is 703-305-3900.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for Page 13 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 703-305-3900 (toll free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, please call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Kuen S. Lu,

Patent Examiner, Art Unit 2167

May 12, 2007

JOHN COTTINGHAM
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

Interview Summary	Application No.	Applicant(s)	
	10/868,434	CARR ET AL.	
	Examiner	Art Unit	
	Kuen S. Lu	2167	

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

(1) *Kuen S. Lu*. (3)_____.

(2) *Mr. Ken J. Koestner (Registration 33,004)*. (4)_____.

Date of Interview: *11 May 2007*.

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

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

Claim(s) discussed: *1-40*.

Identification of prior art discussed: *n/a*.

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

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: *Applicant agreed to an Examiner's Amendment to amend claims 1, 2, 4, 22-28, 33, 36 and 40. Examiner agreed to conduct an update search on prior art and submit, pending update search result, the application for allowance review and approval.*.

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

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

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

Examiner's signature, if required

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 63 of 302

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.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 64 of 302

Notice of References Cited	Application/Control No. 10/868,434	Applicant(s)/Patent Under Reexamination CARR ET AL.	
	Examiner Kuen S. Lu	Art Unit 2167	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,088,717	07-2000	Reed et al.	709/201
*	B	US-2005/0021354	01-2005	Brendle et al.	705/001
*	C	US-2005/0091276	04-2005	Brunswig et al.	707/104.1
*	D	US-7,062,500	06-2006	Hall et al.	707/102
*	E	US-2005/0021998	01-2005	Fiedler et al.	713/200
*	F	US-2005/0044165	02-2005	O'Farrell et al.	709/213
*	G	US-2004/0045014	03-2004	Radhakrishnan, Rakesh	719/328
	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.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 65 of 302

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10868434	CARR ET AL.
	Examiner	Art Unit
	Lu, Kuen S	2167

ORIGINAL		INTERNATIONAL CLASSIFICATION												
CLASS	SUBCLASS	CLAIMED					NON-CLAIMED							
707	104.1	G	0	6	F	17 / 00 (2006.01.01)								
CROSS REFERENCE(S)		G	0	6	F	7 / 00 (2006.01.01)								
		G	0	6	F	15 / 16 (2006.01.01)								

CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)				
707	101	102			
709	201				

Lu, Kuen S /K.S.Lu/ (Assistant Examiner)	5/12/2007 (Date)	/JOHN COTTINGHAM/ SUPERVISORY PATENT EXAMINER TECHNOLOGY CENTER 2100 (Primary Examiner)	Total Claims Allowed: 37
/A White/ (Legal Instruments Examiner)	5/13/07 (Date)	(Date) 5/14/07	O.G. Print Claim(s): 1 / O.G. Print Figure: 3A

U.S. Patent and Trademark Office

Part of Paper No. 10868434-051107

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 66 of 302

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10868434	CARR ET AL.
	Examiner	Art Unit
	Lu, Kuen S	2167

✓	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	05/12/2007								
1	1	=								
2	2	=								
3	3	=								
4	4	=								
5	5	=								
6	6	=								
7	7	=								
8	8	=								
9	9	=								
10	10	=								
11	11	=								
12	12	=								
13	13	=								
14	14	=								
15	15	=								
16	16	=								
17	17	=								
18	18	=								
19	19	=								
20	20	=								
21	21	=								
22	22	=								
23	23	=								
24	24	=								
25	25	=								
26	26	=								
27	27	=								
28	28	=								
29	29	=								
30	30	=								
31	31	=								
32	32	=								
33	33	=								
34	34	=								
35	35	=								
36	36	=								

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 67 of 302

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10868434	CARR ET AL.
	Examiner	Art Unit
	Lu, Kuen S	2167

✓	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	05/12/2007								
	37	-								
	38	-								
	39	-								
37	40	=								

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 68 of 302

Search Notes	Application/Control No.	Applicant(s)/Patent under Reexamination	
	10/868,434	CARR ET AL.	
	Examiner	Art Unit	
	Kuen S. Lu	2167	

SEARCHED			
Class	Subclass	Date	Examiner
707	104.1	5/12/2007	KLU
705	1	5/12/2007	KLU
709	201	5/12/2007	KLU
707	100	5/12/2007	KLU
707	102	5/12/2007	KLU
707	101	5/12/2007	KLU

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner
707	**104.1**	**5/12/2007**	**KLU**
707	102, 101	5/12/2007	KLU
707	100, 201	5/12/2007	KLU
709/201 705/1		5/12/2007	KLU

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	DATE	EXMR
consult with Hung Pham, Examiner of Art Unit 2168, concerning potential 35 USC 101 issues	11/25/2006	KLU
EAST(11/25-28/2006)	5/12/2007	KLU
NPL (google.com, IEEE, ACM)	5/12/2007	KLU
case reviewed by Mr. John Cottingham, Supervisory Primary Examiner, Art Unit 2167	5/14/2007	KLU

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. 4452

SERIAL NUMBER	FILING OR 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
10/868,434	06/14/2004 RULE	707	2167	200400406-1

APPLICANTS

Steven R. Carr, San Jose, CA;
Venkatakrishnan Muthuswamy, Cupertino, CA;
Tudor Har, Santa Clara, CA;

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

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

IF REQUIRED, FOREIGN FILING LICENSE GRANTED
** 08/12/2004

Foreign Priority claimed	STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
☐ yes ☑ no	CA	24	39	8
35 USC 119 (a-d) conditions met ☐ yes ☑ no ☐ Met after Allowance				
Verified and Acknowledged Examiner's Signature Initials				

ADDRESS
22879

TITLE
Data services handler

FILING FEE RECEIVED	FEES: Authority has been given in Paper
1542	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

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 70 of 302

Terms used
assembler disassembler polymorphic transform metadata

Found **1** of **201,062**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 1 of 1

Relevance scale

1 Joeq: a virtual machine and compiler infrastructure

John Whaley

June 2003 **Proceedings of the 2003 workshop on Interpreters, virtual machines and emulators IVME '03**

Publisher: ACM Press

Full text available: pdf(206.13 KB) Additional Information: full citation, abstract, references, citings, index terms

Joeq is a virtual machine and compiler infrastructure designed to facilitate research in virtual machine technologies such as Just-In-Time and Ahead-Of-Time compilation, advanced garbage collection techniques, distributed computation, sophisticated scheduling algorithms, and advanced run time techniques. Joeq is entirely implemented in Java, leading to reliability, portability, maintainability, and efficiency. It is also language-independent, so code from any supported language can be seamlessly ...

Results 1 - 1 of 1

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

EAST Search History

10/868,434

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	2	("20050278270").PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/12 12:07
L2	0	("polymorphicadjmeta$1data").PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/12 12:07
L3	2	polymorphic adj meta$1data	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L4	4	polymorphic with meta$1data	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L5	202	polymorphic and meta$1data	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L6	21	polymorphic and meta$1data and (security with model)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:18
L7	18	polymorphic and meta$1data and (security with model) and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L8	18	polymorphic and meta$1data and (security with model) and @ad < "20040614" and interface	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L9	0	polymorphic and meta$1data and (security with model) and @ad < "20040614" and interface and (data with intergrity)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 72 of 302

EAST Search History

L10	7	polymorphic and meta$1data and (security with model) and @ad < "20040614" and interface and (data with integrity)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L11	73	mid$4ware and meta$1data and (security with model) and @ad < "20040614" and interface and (data with integrity)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L12	2	(mid$4ware with service with manager) and meta$1data and (security with model) and @ad < "20040614" and interface and (data with integrity)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L13	0	(service with manager with meta$1data) and (security with model) and @ad < "20040614" and interface and (data with integrity)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L14	3	(service with manager with application with (database or (data adj source))) and meta$1data and (security with model) and @ad < "20040614" and interface and (data with integrity)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L15	61	(service with manager with application with (database or (data adj source))) and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L16	0	(service with manager with application with (database or (data adj source))) and (meta$1data with types with attributes) and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L17	47	(service with manager with application with (database or (data adj source))) and (meta$1data and types and attributes and class and object) and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L18	0	(service with manager with application with (database or (data adj source))) and (meta$1data and types and attributes and class and object) and @ad < "20040614" and polymorhic	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 73 of 302

EAST Search History

L19	0	(service with manager with application with (database or (data adj source))) and (meta$1data and types and attributes and class) and @ad < "20040614" and (polymorphic with object)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L20	0	(service with manager with application with (database or (data adj source))) and (meta$1data and types and attributes and class) and @ad < "20040614" and (polymorphic same object)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L21	11	(service with manager with application with (database or (data adj source))) and (meta$1data and types and attributes and class and object) and @ad < "20040614" and polymorphic	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L22	1	(meta$1data or polymorphic) and "20050021354"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L23	0	("6088717.pn.").PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/12 12:07
L24	2	"6088717".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L25	2	"20050027871"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L26	4	"20050021354" or "6088717".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L27	4	707/104.1.ccls. and (service with manager with application with (database or (data adj source))) and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 74 of 302

EAST Search History

L28	4	705/1.ccls. and (service with manager with application with (database or (data adj source))) and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L29	2	707/100.ccls. and (service with manager with application with (database or (data adj source))) and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:17
L30	2	709/201.ccls. and (service with manager with application with (database or (data adj source))) and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L31	2	"20050278270"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/12 12:07
L32	4	"20020073399" or "6088717".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/12 12:07
L33	6	"2002007399" or "6088717".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/12 12:07
L34	0	("20020004819"or"20020123993"). PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/12 12:07
L35	2	("20020004819").PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/12 12:07

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 75 of 302

EAST Search History

L36	4	"20020004819" or "20020123993"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/12 12:07
L37	1028	language adj commands	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/12 12:07
L38	4	native adj language adj commands	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/12 12:07
L39	0	native adj language adj script	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/12 12:07
L40	14	(language adj commands) with XML	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/12 12:07
L41	3	"6810429".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/12 12:07
L42	4	"20020004819" or "20020123993"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/12 12:07

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 76 of 302

EAST Search History

L43	2	"20050278270"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/12 12:07
L44	2	"20050071737"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:07
L45	2	"6732095".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/12 12:07
L46	0	707/101.ccls. and (service with manager with application with (database or (data adj source))) and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:17
L47	0	707/101.ccls. and polymorphic and meta$1data and (security with model)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:18
L48	0	707/100.ccls. and polymorphic and meta$1data and (security with model)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:18
L49	12	707/100.ccls. and polymorphic and meta$1data	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:19
L50	10	707/100.ccls. and polymorphic and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:19
L51	3	707/101.ccls. and polymorphic and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:19

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 77 of 302

EAST Search History

L52	13	707/102.ccls. and polymorphic and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:19
L53	11	707/104.1.ccls. and polymorphic and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:19
L54	6	707/103R.ccls. and polymorphic and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:20
L55	7	709/201.ccls. and polymorphic and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/12 12:20
S1	2	("20050278270").PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/08 19:10
S2	0	("polymorphicadjmeta$1data").PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/25 16:50
S3	2	polymorphic adj meta$1data	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/25 16:51
S4	2	polymorphic with meta$1data	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/25 16:51
S5	183	polymorphic and meta$1data	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/25 16:51
S6	19	polymorphic and meta$1data and (security with model)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/25 16:52

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 78 of 302

EAST Search History

S7	16	polymorphic and meta$1data and (security with model) and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/25 16:52
S8	16	polymorphic and meta$1data and (security with model) and @ad < "20040614" and interface	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/25 16:53
S9	0	polymorphic and meta$1data and (security with model) and @ad < "20040614" and interface and (data with intergrity)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/25 16:53
S10	6	polymorphic and meta$1data and (security with model) and @ad < "20040614" and interface and (data with integrity)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 10:42
S11	63	mid$4ware and meta$1data and (security with model) and @ad < "20040614" and interface and (data with integrity)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 10:42
S12	1	(mid$4ware with service with manager) and meta$1data and (security with model) and @ad < "20040614" and interface and (data with integrity)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 10:47
S13	0	(service with manager with meta$1data) and (security with model) and @ad < "20040614" and interface and (data with integrity)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 10:47
S14	3	(service with manager with application with (database or (data adj source))) and meta$1data and (security with model) and @ad < "20040614" and interface and (data with integrity)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 12:46
S15	51	(service with manager with application with (database or (data adj source))) and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/28 15:43

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 79 of 302

EAST Search History

S16	0	(service with manager with application with (database or (data adj source))) and (meta$1data with types with attributes) and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 12:47
S17	37	(service with manager with application with (database or (data adj source))) and (meta$1data and types and attributes and class and object) and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 12:48
S18	0	(service with manager with application with (database or (data adj source))) and (meta$1data and types and attributes and class and object) and @ad < "20040614" and polymorhic	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 12:49
S19	11	(service with manager with application with (database or (data adj source))) and (meta$1data and types and attributes and class and object) and @ad < "20040614" and polymorphic	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 14:27
S20	0	(service with manager with application with (database or (data adj source))) and (meta$1data and types and attributes and class) and @ad < "20040614" and (polymorphic with object)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 12:49
S21	0	(service with manager with application with (database or (data adj source))) and (meta$1data and types and attributes and class) and @ad < "20040614" and (polymorphic same object)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 12:49
S22	1	(meta$1data or polymorphic) and "20050021354"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 14:27
S23	0	("6088717.pn.").PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/26 14:36
S24	2	"6088717".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 14:36

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 80 of 302

EAST Search History

S25	2	"20050027871"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 14:38
S26	4	"20050021354" or "6088717".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/27 10:09
S27	4	707/104.1.ccls. and (service with manager with application with (database or (data adj source))) and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/28 15:44
S28	4	705/1.ccls. and (service with manager with application with (database or (data adj source))) and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/28 15:43
S29	2	707/100.ccls. and (service with manager with application with (database or (data adj source))) and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/28 15:44
S30	1	709/201.ccls. and (service with manager with application with (database or (data adj source))) and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/28 15:44
S31	2	"20050278270"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/08 13:12
S32	6	"2002007399" or "6088717".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/08 16:14
S33	4	"20020073399" or "6088717".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/08 16:14

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 81 of 302

EAST Search History

S34	2	("20020004819").PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/08 19:10
S35	0	("20020004819"or"20020123993"). PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/08 17:40
S37	4	"20020004819" or "20020123993"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/08 17:50
S38	0	native adj language adj script	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/08 17:51
S39	4	native adj language adj commands	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/08 17:53
S40	1027	language adj commands	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/08 17:53
S41	14	(language adj commands) with XML	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/08 18:06

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 82 of 302

EAST Search History

S42	3	"6810429".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/09 17:29
S43	2	"20050278270"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/08 19:10
S44	4	"20020004819" or "20020123993"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/08 19:20
S45	2	"20050071737"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/05/09 13:28
S46	2	"6732095".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/05/09 17:29

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 83 of 302

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

+assembler +disassembler +polymorphic +consume +transfor SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used
assembler disassembler polymorphic consume transform security model metadata

Found **1** of **201,062**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 1 of 1

Relevance scale

1 Joeq: a virtual machine and compiler infrastructure

John Whaley

June 2003 **Proceedings of the 2003 workshop on Interpreters, virtual machines and emulators IVME '03**

Publisher: ACM Press

Full text available: pdf(206.13 KB) Additional Information: full citation, abstract, references, citings, index terms

Joeq is a virtual machine and compiler infrastructure designed to facilitate research in virtual machine technologies such as Just-In-Time and Ahead-Of-Time compilation, advanced garbage collection techniques, distributed computation, sophisticated scheduling algorithms, and advanced run time techniques. Joeq is entirely implemented in Java, leading to reliability, portability, maintainability, and efficiency. It is also language-independent, so code from any supported language can be seamlessly ...

Results 1 - 1 of 1

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 84 of 302

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

+assembler +disassembler +polymorphic +consume +transfor SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used
assembler disassembler polymorphic consume transform metadata

Found **1** of **201,062**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 1 of 1

Relevance scale

1 Joeq: a virtual machine and compiler infrastructure

John Whaley
June 2003 **Proceedings of the 2003 workshop on Interpreters, virtual machines and emulators IVME '03**
Publisher: ACM Press

Full text available: pdf(206.13 KB) Additional Information: full citation, abstract, references, citings, index terms

Joeq is a virtual machine and compiler infrastructure designed to facilitate research in virtual machine technologies such as Just-In-Time and Ahead-Of-Time compilation, advanced garbage collection techniques, distributed computation, sophisticated scheduling algorithms, and advanced run time techniques. Joeq is entirely implemented in Java, leading to reliability, portability, maintainability, and efficiency. It is also language-independent, so code from any supported language can be seamlessly ...

Results 1 - 1 of 1

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 85 of 302

PORTAL
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: The ACM Digital Library The Guide

+assembler +disassembler +polymorphic +convert +metadata SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used
assembler disassembler polymorphic convert metadata

Found **1** of **201,062**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 1 of 1

Relevance scale

1 Joeq: a virtual machine and compiler infrastructure

John Whaley
June 2003 **Proceedings of the 2003 workshop on Interpreters, virtual machines and emulators IVME '03**
Publisher: ACM Press

Full text available: pdf(206.13 KB) Additional Information: full citation, abstract, references, citings, index terms

Joeq is a virtual machine and compiler infrastructure designed to facilitate research in virtual machine technologies such as Just-In-Time and Ahead-Of-Time compilation, advanced garbage collection techniques, distributed computation, sophisticated scheduling algorithms, and advanced run time techniques. Joeq is entirely implemented in Java, leading to reliability, portability, maintainability, and efficiency. It is also language-independent, so code from any supported language can be seamlessly ...

Results 1 - 1 of 1

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

PORTAL
USPTO
Subscribe (Full Service) Register (Limited Service, Free) Login
Search: The ACM Digital Library The Guide
+assembler +disassembler +polymorphic +metadata
SEARCH
THE ACM DIGITAL LIBRARY
Feedback Report a problem Satisfaction survey

Terms used **assembler disassembler polymorphic metadata** Found **1** of **201,062**

Sort results by relevance
Display results expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 1 of 1

Relevance scale

1 Joeq: a virtual machine and compiler infrastructure

John Whaley

June 2003 **Proceedings of the 2003 workshop on Interpreters, virtual machines and emulators IVME '03**

Publisher: ACM Press

Full text available: pdf(206.13 KB) Additional Information: full citation, abstract, references, citings, index terms

Joeq is a virtual machine and compiler infrastructure designed to facilitate research in virtual machine technologies such as Just-In-Time and Ahead-Of-Time compilation, advanced garbage collection techniques, distributed computation, sophisticated scheduling algorithms, and advanced run time techniques. Joeq is entirely implemented in Java, leading to reliability, portability, maintainability, and efficiency. It is also language-independent, so code from any supported language can be seamlessly ...

Results 1 - 1 of 1

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.
Terms of Usage Privacy Policy Code of Ethics Contact Us

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player



Results for "((polymorphic <in>metadata) <and> (metadata<in>metadata))" e-mail
Your search matched 0 documents.
A maximum of 100 results are displayed, 25 to a page, sorted by Relevance in Descending order.

» Search Options

View Session History

New Search

Modify Search

((polymorphic <in>metadata) <and> (metadata<in>metadata)) Search

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IET JNL	IET Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IET CNF	IET Conference Proceeding
IEEE STD	IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer to the Help pages if you need assistan search.

Indexed by Inspec®

© Copyright 2006 IEEE –

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 88 of 302

IEEE Xplore® RELEASE 2.3

Home | Login | Logout | Access Information | Alerts |

Welcome United States Patent and Trademark Office

☐ Search Results BROWSE SEARCH IEEE XPLORE GUIDE

Results for "((polymorphic<in>metadata) <and> (assembler<in>metadata))" e-mail
Your search matched **0** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» Search Options

View Session History

New Search

Modify Search

((polymorphic<in>metadata) <and> (assembler<in>metadata)) Search

☐ Check to search only within this results set

Display Format: ◉ Citation ○ Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IET JNL	IET Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IET CNF	IET Conference Proceeding
IEEE STD	IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer to the Help pages if you need assistan search.

Help Contact Us Privacy &

Indexed by Inspec®

© Copyright 2006 IEEE –

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 89 of 302

kuen_lu@hotmail.com | Web History | My Account | Sign out

Google Web Images Video News Maps more »
"polymorphic metadata" Search Advanced Search Preferences
New! View and manage your web history

Web Results **1** - **13** of **13** for **"polymorphic metadata"**. (**0.15** seconds)

Boost mailing page: Re: [boost] Re: compile-time functions interface
Types are the only fully **polymorphic metadata** because they > can wrap anything else (well, as soon as we get template typedefs they ...
lists.boost.org/Archives/boost/2003/02/43602.php - 19k -
Cached - Similar pages - Note this

THE BOOST MPL LIBRARY: Details
The most basic way to formulate a compile-time function so that it can be treated as **polymorphic metadata**; that is, as a type. A metafunction class is a ...
www.boost.org/libs/mpl/doc/tutorial/details.html - 8k -
Cached - Similar pages - Note this

[PPT] The Boost Metaprogramming Library
File Format: Microsoft Powerpoint - View as HTML
Templates are not **polymorphic metadata** (types)! ERROR. copyright 2003 David Abrahams. *. 26. Metafunction Classes. Applying familiar type-wrapper idiom: ...
www.boost-consulting.com/writing/ACCU_MPL_slides.ppt - Similar pages - Note this

[PDF] 3 [Please insert PrerenderUnicode{ } into preamble] A Deeper Look ...
File Format: PDF/Adobe Acrobat - View as HTML
can be treated as **polymorphic metadata**; that is, as a type. A metafunction class is a. class with a nested metafunction called apply. ...
www.boost-consulting.com/tmpbook/metafunctions.pdf - Similar pages - Note this
[More results from www.boost-consulting.com]

Caml Weekly News
There are also problems with typing, e.g. sets of related functions that handle **polymorphic metadata** of the same type between each of the functions. ...
alan.petitepomme.net/cwn/2005.10.11.html - 10k - Cached - Similar pages - Note this

Addison-Wesley Professional - 0321227255 - C++ Template ...
As we hinted earlier, the MPL supplies a group of wrapper templates that, like int_, are used to make integer values into **polymorphic metadata**. ...
safari.awprofessional.com/0321227255/ch04 - Similar pages - Note this

[PDF] "binary.hpp" ::type ::value ::value_type <algorithm> <empty.hpp ...
File Format: PDF/Adobe Acrobat - View as HTML
polymorphic metadata, 61. polymorphism, 30–32. definition of, 17. example, 39. function templates and, 196. parametric, 17. static, 17, 196 ...
www.awprofessional.com/content/images/0321227255/index/abrahams_index.pdf - Similar pages - Note this

Data services handler patent invention
... (RTID) that transforms data under direction of **polymorphic metadata** that defines a security model and data integrity rules for application to the data.
www.freshpatents.com/Data-services-handler-dt20051215ptan20050278270.php?type=claims - 32k - Supplemental Result - Cached - Similar pages - Note this

Sponsored Links

Metadata
HP Neoview Platform - Decision Making & Action Taking.
www.hp.com/go/neoview

Metadata Solutions
Gain productivity & agility through metadata management. Free download.
www.informatica.com

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 90 of 302

Data services handler patent invention
[0033] **Polymorphic metadata** can be used to enable extension of the real time information ... **Polymorphic metadata** is metadata that takes many forms, ...
www.freshpatents.com/Data-services-handler-dt20051215ptan20050278270.php?type=description - 210k - Supplemental Result - Cached - Similar pages - Note this
[More results from www.freshpatents.com]

fa.caml | Google Groups
handle **polymorphic metadata** of the same type between each of the function s. > This could be implemented using functors, records of functions or objects ...
groups.google.com/group/fa.caml/tree/browse_frm/month/2005-10/0e36322c18527a42?rnum=131&_done=%2Fgrou... - 147k - Supplemental Result - Cached - Similar pages - Note this

Re: [boost] compile-time functions interface
Types are the only fully **polymorphic metadata** because they can wrap anything else (well, as soon as we get template typedefs they will be able to wrap ...
www.mail-archive.com/boost@lists.boost.org/msg04466.html - 11k - Supplemental Result - Cached - Similar pages - Note this

[Boost-docs] Re: Boost Book Status
[Boost-docs] Re: Boost Book Status. David Abrahams Wed, 09 Mar 2005 17:11:37 -0800. David Abrahams <> writes: > I've attached the ...
www.mail-archive.com/boost-docs@lists.sourceforge.net/msg00917.html - 62k - Supplemental Result - Cached - Similar pages - Note this

/vendorsrc/boost/1.32.0/libs/mpl/doc/tutorial/details.html ...
</dt> <dd>The most basic way to formulate a compile-time function so that it can be treated as **polymorphic metadata**; that is, as a type. ...
code.crt.realtors.org/projects/librets/file/vendorsrc/boost/1.32.0/libs/mpl/doc/tutorial/details.html?rev=533 - 34k - Supplemental Result - Cached - Similar pages - Note this

In order to show you the most relevant results, we have omitted some entries very similar to the 13 already displayed.
If you like, you can repeat the search with the omitted results included.

Try Google Desktop: search your computer as easily as you search the web.

"polymorphic metadata" Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Programs - Business Solutions - About Google

©2007 Google

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 91 of 302

kuen_lu@hotmail.com | Web History | My Account | Sign out

Google

Web Images Video News Maps more »

assembler disassembler polymorphic supply s Search Advanced Search Preferences

New! View and manage your web history

Web Results **1** - **43** of **43** for **assembler disassembler polymorphic supply stateful tables director consum**

Javvin Networking Information and Resources
Biometric Technology and Its Application to Information **Security** ... Bell-LaPadula **Security Model** ... FRAD: Frame Relay **Assembler/Disassembler** ...
www.javvin.com/sitemap.html - 780k - Cached - Similar pages - Note this

Asia/Pacific Computer Services - links about Computer Programming ...
Compiler / Decompiler / **Assembler** / **Disassembler** / **Assembly** / Interpreter ... The core **model** repository is an implementation of the Java **Metadata** Interface ...
asiapac.com.au/Links/Programming.htm - 595k - Cached - Similar pages - Note this

[PDF] The Ultimate Computer Acronyms Archive
File Format: PDF/Adobe Acrobat
DTM: Document **Table Model**. Interface to a document **model** designed specifically for the ... **Assembler/Disassembler** (FRAD) and the service provider ...
www.acronyms.ch/files/Acronyms.Letter.Duplex.pdf - Similar pages - Note this

Science Library Pad: Presentation Notes
Unified way to **model** and interact with **stateful** Web Services. Peer-to-Peer Computing. Summary ... NSDL is not a content library, it is a **metadata** library ...
scilib.typepad.com/science_library_pad/presentation_notes/index.html - 365k - Cached - Similar pages - Note this

> Science Library Pad: Conference
> Unified way to **model** and interact with **stateful** Web Services ... Head, Data Library and Academic **Director** of the Research Data Centre University of Alberta ...
> scilib.typepad.com/science_library_pad/conference/index.html - 977k - Cached - Similar pages - Note this

[DOC] 1
File Format: Microsoft Word - View as HTML
The **security** of a component **assembly** or integrated system emerges from the behaviors of all of ... By automating the **transformation** from UML **model** to a PSM, ...
standards.computer.org/sesc/s2esc_excom/minutes/2006-08/DHS%20Lifecycle_DraftV11-072806.doc - Similar pages - Note this

[PDF] Oracle9 Application Server
File Format: PDF/Adobe Acrobat
model, naming and directory lookup, or remote connectivity **model**. ... **metadata** repository for storing its **tables** and configuration. Each application server ...
download-east.oracle.com/docs/cd/A97329_01/core.902/a95926.pdf - Similar pages - Note this

[PDF] Oracle Application Server 10g Concepts
File Format: PDF/Adobe Acrobat
The TopLink application **metadata model** is based around the OracleAS TopLink ... State replications and routing: For **stateful** client requests, client state ...
docs.huihoo.com/oracle/docs/B14099_19/core.1012/b13994.pdf - Similar pages - Note this

www.ridopark.info > Home

Resource File: Threat **Model** Your **Security** Risks ... the MSIL **disassembler**, are

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 92 of 302

discussed—as well as how **metadata**, manifests, and assemblies simplify ...
www.ridopark.info/ - 619k - Cached - Similar pages - Note this

[PDF] Oracle Application Server 10g Concepts
File Format: PDF/Adobe Acrobat
The TopLink application **metadata model** is based around the OracleAS TopLink ...
stateful requests are forwarded to the OC4J instance that served the ...
otndnld.oracle.co.jp/document/products/as10g/904/doc_v7/core.904/B10375_02.pdf -
Similar pages - Note this

Sa-Sm
identifying **polymorphic** sites and generating a **table** summarizing the ... This is a producer-**consumer model** in which blocks of loop iterations called ...
stommel.tamu.edu/~baum/linuxlist/linuxlist/node43.html - 405k -
Cached - Similar pages - Note this

[PDF] The Ultimate Computer Acronyms Archive
File Format: PDF/Adobe Acrobat
DTM: Document **Table Model**. Interface to a document **model** designed specifically for the ... The interface between the Frame Relay **Assembler/Disassembler** ...
www.pctaxpro.com/pcrepair/files/Acronyms.pdf - Similar pages - Note this

[PDF] The Gartner Glossary
File Format: PDF/Adobe Acrobat
assembler/disassembler (FRAD) on the receiving end of a transmission backward to ... in a **metadata**-independent manner. See **transformation**. ...
www.gartner.com/6_help/glossary/Gartner_IT_Glossary.pdf - Similar pages - Note this

[PDF] The CISSP Prep Guide Gold Edition
File Format: PDF/Adobe Acrobat
A **disassembler**. reverses the function of an **assembler** by translating machine ... **Supplies** and equipment. Paper, forms, HVAC, or specialized **security** ...
www.team509.com/download/docs/**security**/management/
Wiley.The.CISSP.Prep.Guide.Gold.Edition.eBook-kB.pdf - Similar pages - Note this

Java performance tuning tips
Don't **model** a shared cache or any shared resource as a **stateful** session bean. ...
Security checks **consume** CPU resources. You will get better performance if ...
www.javaperformancetuning.com/tips/rawtips.shtml - 419k -
Cached - Similar pages - Note this

[PDF] Oracle® Database
File Format: PDF/Adobe Acrobat
security model within the database, certain differences manifest themselves. ... sequence of **transformation** without a **table** holding intermediate results. ...
www.stanford.edu/dept/itss/docs/oracle/10g/java.101/b12021.pdf - Similar pages - Note this

> [PDF] Oracle® Database
> File Format: PDF/Adobe Acrobat
> Java has inheritance, **polymorphism**, and component **models** for developing ... The **consumer** function or SQL statement uses the **TABLE** keyword to treat the ...
> www.stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10795.pdf -
> Similar pages - Note this

[PDF] Liste von Abkurzungen

File Format: PDF/Adobe Acrobat
Membrane Electrode **Assembly** (Fuel Cell). MED. Maximum Entropy Distribution. MEDEA. Microelectronics Development for European Applications. MEET Man. **Models** ...

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 93 of 302

www.dpi.physik.uni-goettingen.de/~guicking/abk-allg.pdf - Similar pages - Note this

Security Taxonomy and Glossary - Anne & Lynn Wheeler
*-property: IncludedBy:Bell-LaPadula **security model**,: PreferredFor:star (*) property, ...
open systems **security**, packet **assembly** and disassembly, ...
www.garlic.com/~lynn/secure.htm - 1k - Cached - Similar pages - Note this

FreeBSD Ports Collection Index
flasm-1.52 -- Command line **assembler/disassembler** of flash actionscript bytecode ...
portaudit-0.5.11 -- Checks installed ports against a list of **security** ...
www.ee.freebsd.org/ports/master-index.html - Similar pages - Note this

JOS Statistics - Recent Comments
Tue 20 Jul | Eric Debois | I cant think of a single electronic **consumer** device that didnt ...
Why cant HP provide a **table** with all **models**, with simple rows, ...
www.usabilitymustdie.com/jos/WW_Weekly_Comments_25_Jul_2004.html -
Similar pages - Note this

> JOS Statistics - Recent Comments
> Fri 16 Apr | Nam | Regarding compiling the **metadata**; how do you take care ... Does the ShareWare **model** is still viable in 2004 | Sun 11 Apr | Linus Ericsson ...
> www.usabilitymustdie.com/jos/WW_Weekly_Comments_18_Apr_2004.html - 951k -
> Cached - Similar pages - Note this

Hibernate In Action - Manning - 2005
We briefly discuss Hibernate's fully **polymorphic** behavior as well. ... A view may be used to **transform** the **table**-per-subclass **model** into the much simpler ...
www.scribd.com/doc/23334/Hibernate-In-Action-Manning-2005 - 908k -
Cached - Similar pages - Note this

> Wrox Press - Professional ASP NET Programming Part 3
> NET **Assembly**. In this section we'll discuss the programming **model**. ... which provides methods to convert COM classes and interfaces into **assembly meta data**. ...
> www.scribd.com/doc/2826/Wrox-Press-Professional-ASP-NET-Programming-Part-3 - 688k -
> Cached - Similar pages - Note this
> [More results from www.scribd.com]

bgran » Sun
So I **consume** insane amounts of caffeine to stay alert. ... Since I'm also going to do some software cracking I will run Softice and IDA code **disassembler**. ...
bgran.net/?cat=9 - 321k - Cached - Similar pages - Note this

words by the number of messages they appear
listDirectories directoryLister disapprove **disassembler** 3 disasters disclaimer ... spin sponsor sponsored spurious stare **stateful** stress 16 stupidly subol ...
twistedmatrix.com/users/acapnotic/wip/wordcount-3plus.txt - 284k -
Cached - Similar pages - Note this

[PDF] .NET Tutorialfor Beginners
File Format: PDF/Adobe Acrobat
The **assembly** manifest contains the **assembly's metadata** and it also ... To visualize the IL code Microsoft provides a **disassembler** tool ...
download.microsoft.com/.../e/7/8e725d96-7ec3-498b-9fa7-86779aed101f/dotNET%20Tutorial%20for%20Beginners.pdf - Similar pages - Note this

[Linux Mall] Sempre conectada com você!- [Translate this page]
Obtaining Database **Metadata** Chapter 8. **Tables** and Indexes Section 8.1. ... Platform Specific **Model** - PSM 2. **Model Transformation** 3. Padrões MDA 3. ...

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 94 of 302

www.linuxmall.com.br/index.php?page=categ/flypage_info&print=1&product_id= -
Similar pages - Note this

[PDF] Seminar System Modeling
File Format: PDF/Adobe Acrobat
provides a **model** for the **assembly** of business ... also imaginable to use a **stateful** session bean instead – ... **Disassembler** that disassembles a multi- ...
wendtstud1.hpi.uni-potsdam.de/sysmod-seminar/SS2006/SysmodSeminar2006.pdf -
Similar pages - Note this

RPM shipped by Mandriva
perl-Parse-RecDescent-**Consumer**-1.03-1mdv2007.0.noarch, Parse::RecDescent::**Consumer** ... Perl module that **supplies** object methods for interacting with GnuPG ...
rpmfind.net/linux/RPM/mandriva/devel/cooker/sparc64/media/contrib/Mandriva.html - 977k -
Cached - Similar pages - Note this

> RPM of Distribution Mandriva Linux
> avra-1.1.0-1mdv2007.1.sparc, An AVR **assembler**, Mandriva devel cooker for sparc ... file manipulation tool and **disassembler**, Mandriva devel cooker for sparc ...
> rpmfind.net/linux/RPM/mandriva/devel/cooker/sparc/Mandriva_Linux.html -
> Similar pages - Note this
> [More results from rpmfind.net]

SpeedyFeed.com : Social Syndication - robl's SpeedyFeeds
[random guy] pointed out that Canon firmware hacking is still alive and well, but it's the smaller **consumer models** that are getting the attention. ...
www.speedyfeed.com/robl/ - 977k - Cached - Similar pages - Note this

Version 0.10.1 ========== * lots of small fixes Translations ...
Added ability to register 'device event listeners' which can **consume** device ... Built-in support for **assembly** * aclocal gives error if `AM_' macro not found ...
ozlabs.org/~jk/projects/lca2007-hackfest/q3/data.in - Similar pages - Note this

こんな毎日・・・ Recent_SourceForge_News- [Translate this page]
SISCweb is a framework to facilitate writing **stateful** Scheme web applications ... Graph **transformation**, **model transformation**, object-oriented verification, ...
machine.homeunix.org/weblog/archives/sf_news_2005-06-21.html -
Similar pages - Note this

View Resumes
Only Session Beans are used (both **Stateful** and Stateless). The front end is done in XSQL ... Symbol **Table** program for a compiler. 5. **Security** Issues report. ...
www.teleworkrecruiting.com/resume-detail.php?category=IT+and+Computer+Programming - Similar pages - Note this

[PDF] Wielka encyklopedia komputerów
File Format: PDF/Adobe Acrobat
cross **assembler**, 106. cross compiler, 106. cross media, 106 ... obsługują te **model** „wiele do wielu". Obiektowe bazy danych są ...
czytelnia.reporter.pl/rozdzialy/wencko.pdf - Similar pages - Note this

[PDF] Oracle® Database
File Format: PDF/Adobe Acrobat
Java has inheritance, **polymorphism**, and component **models** for developing ... need to store the data in **tables** between each **transformation** stage. ...
www.rdbprime.com/Oracle/Oracle_Docs/Oracle10gDB_Server/appdev.101/b10795.pdf -
Similar pages - Note this

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 95 of 302

[PDF] Oracle® Database
File Format: PDF/Adobe Acrobat
Java has inheritance, **polymorphism**, and component **models** for developing ... **tables** between each **transformation** stage. These functions can accept and return ...
www.sme-solution.com/Oracle/B19306_01/appdev.102/b14251.pdf -
Similar pages - Note this

dict200k txt
... **assembler Assembler** assemblies Assemblies assembling Assembling **assembly** ... Disarticulation Disarticulator **disassemble disassembler** disassembly ...
www.cs.bsu.edu/gwen/peb/cs232/Projects/SpellCheck/dict200k.txt -
Similar pages - Note this

@requires@ORBit-devel@automake1.4@automake1.7@db1-devel@gettext ...
File Format: Unrecognized - View as HTML
... @summary@Libextractor library used to extract **meta-data** from files ... and **disassembler** @info@biew-5.6.4-1mdv2008.0.src@0@628332@File tools ...
ftp.cica.es/Mandrake-devel/cooker/i586/media/media_info/synthesis.hdlist_contrib.src.cz -
Similar pages - Note this

all my del icio us posts at someplace
Working **Model** Steam Engines, Machine Tool Kits, **Model Supplies**, ... r / json html **transform** cool xslt xml example code javascript del.icio.us jsont minimal ...
mshook.python-hosting.com/delicious/all.html - Similar pages - Note this

1017 Internet 961 protocol 932 Unix 928 TCP/IP 911 search engine ...
... 15 Parallel Sysplex 15 Packet **Assembler/Disassembler** 15 PRECCX 15 PD 15 PCI ... 1 **stateful** 1 stated function 1 state-free 1 state **table** 1 state space ...
www.kambarys.lt/test.php - 854k - Cached - Similar pages - Note this

aux ciar org ttk lemma ttkciar 01 txt
... stars 1294 tungsten 1294 **table** 1294 laser 1294 directory 1294 decoration ... 51 bhorsepic 51 beats 51 atsushi 51 **assembler** 51 arsenic 51 anita 51 alumni ...
aux.ciar.org/ttk/lemma.ttkciar.01.txt - Similar pages - Note this

In order to show you the most relevant results, we have omitted some entries very similar to the 43 already displayed.
If you like, you can repeat the search with the omitted results included.

---

assembler disassembler polymorphi Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

---

Google Home - Advertising Programs - Business Solutions - About Google

©2007 Google

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 96 of 302

kuen_lu@hotmail.com | Web History | My Account | Sign out

Google

Web Images Video News Maps more »

assembler disassembler polymorphic supply s Search Advanced Search Preferences

New! View and manage your web history

Web Results **1** - **46** of **46** for **assembler disassembler polymorphic supply stateless tables director consu**

Sandline Training Demo Page
Chapter Four - Databases And **Tables** Creating A Database Recovery **Model** ... **Assembly** Parts **Metadata** Strong Name & Reflection **Security** Overview ...
www.sandline-training.co.uk/list-cbt.htm - 355k - Cached - Similar pages - Note this

Asia/Pacific Computer Services - links about Computer Programming ...
Compiler / Decompiler / **Assembler** / **Disassembler** / **Assembly** / Interpreter ... The core **model** repository is an implementation of the Java **Metadata** Interface ...
asiapac.com.au/Links/Programming.htm - 595k - Cached - Similar pages - Note this

[PDF] The Ultimate Computer Acronyms Archive
File Format: PDF/Adobe Acrobat
DTM: Document **Table Model**. Interface to a document **model** designed specifically for the ... **Assembler/Disassembler** (FRAD) and the service provider ...
www.acronyms.ch/files/Acronyms.Letter.Duplex.pdf - Similar pages - Note this

[PDF] Improving Software **Security** with Precise Static and Runtime Analysis
File Format: PDF/Adobe Acrobat
Security label **polymorphism** allows for code that is generic with ... to merge **security** automata directly into binary code at the x86 **assembler** and Java ...
research.microsoft.com/~livshits/papers/pdf/thesis.pdf - Similar pages - Note this

[PDF] Oracle9 Application Server
File Format: PDF/Adobe Acrobat
model, naming and directory lookup, or remote connectivity **model**. ... **metadata** repository for storing its **tables** and configuration. Each application server ...
download-east.oracle.com/docs/cd/A97329_01/core.902/a95926.pdf - Similar pages - Note this

[DOC] 1
File Format: Microsoft Word - View as HTML
The **security** of a component **assembly** or integrated system emerges from the behaviors of all of ... By automating the **transformation** from UML **model** to a PSM, ...
standards.computer.org/sesc/s2esc_excom/minutes/2006-08/DHS%20Lifecycle_DraftV11-072806.doc - Similar pages - Note this

[PDF] Oracle Application Server 10g Concepts
File Format: PDF/Adobe Acrobat
The TopLink application **metadata model** is based around the OracleAS TopLink ... Web services include Java classes, **stateless** session EJBs, and **stateless** ...
docs.huihoo.com/oracle/docs/B14099_19/core.1012/b13994.pdf - Similar pages - Note this

www.ridopark.info > Home
Resource File: Threat **Model** Your **Security** Risks ... the MSIL **disassembler**, are discussed—as well as how **metadata**, manifests, and assemblies simplify ...
www.ridopark.info/ - 619k - Cached - Similar pages - Note this

[PDF] The Ultimate Computer Acronyms Archive

File Format: PDF/Adobe Acrobat
DTM: Document **Table Model**. Interface to a document **model** designed specifically for

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 97 of 302

the ... The interface between the Frame Relay **Assembler/Disassembler** ...
www.pctaxpro.com/pcrepair/files/Acronyms.pdf - Similar pages - Note this

[PDF] The Gartner Glossary
File Format: PDF/Adobe Acrobat
assembler/disassembler (FRAD) on the receiving end of a transmission backward to ... in a **metadata**-independent manner. See **transformation**. ...
www.gartner.com/6_help/glossary/Gartner_IT_Glossary.pdf - Similar pages - Note this

计算机与网络词典(コンピュータとネットワーク専門用語の英中対訳)- [Translate this page]
packet **assembler/disassembler**, PAD 封包组合及拆解. packet **assembly** 小包组合 ...
Security Mode 安全模式. **security** Object **Model** 安全对象模型 ...
www.geocities.jp/ps_dictionary/cedic/computer_network.htm - 760k -
Cached - Similar pages - Note this

Java performance tuning tips
A non-static factory method is **polymorphic** and so provides many advantages ... **Security** checks **consume** CPU resources. You will get better performance if you ...
www.javaperformancetuning.com/tips/rawtips.shtml - 419k -
Cached - Similar pages - Note this

[PDF] T E A M F L Y
File Format: PDF/Adobe Acrobat
wide **security model**. Only certain users can run a component, for example. ... **Metadata** is compiled directly into an **assembly** and cannot become separated ...
www.elib.hbi.ir/computer/webdevelopment/pdf/%5B0672322196%5D.Net%20e-Business%20Architecture.pdf - Similar pages - Note this

[PDF] Oracle® Database
File Format: PDF/Adobe Acrobat
security model within the database, certain differences manifest themselves. ... sequence of **transformation** without a **table** holding intermediate results. ...
www.stanford.edu/dept/itss/docs/oracle/10g/java.101/b12021.pdf - Similar pages - Note this

> [PDF] Oracle® Database
> File Format: PDF/Adobe Acrobat
> Java has inheritance, **polymorphism**, and component **models** for developing ... The **consumer** function or SQL statement uses the **TABLE** keyword to treat the ...
> www.stanford.edu/dept/itss/docs/oracle/10g/appdev.101/b10795.pdf -
> Similar pages - Note this

MS Press - Applied .NET Framework Programming (2002)
As for **security**, the .NET Framework includes a new **security model** called code ... **Table** 2-1: Common Definition **Metadata Tables Metadata** Definition **Table** ...
www.scribd.com/doc/1852/MS-Press-Applied-NET-Framework-Programming-2002 - 977k -
Cached - Similar pages - Note this

> Hibernate In Action - Manning - 2005
> We briefly discuss Hibernate's fully **polymorphic** behavior as well. ... A view may be used to **transform** the **table**-per-subclass **model** into the much simpler ...
> www.scribd.com/doc/23334/Hibernate-In-Action-Manning-2005 - 908k -
> Cached - Similar pages - Note this
> [More results from www.scribd.com]

Free On-line Dictionary of Computing FOLDOC is a searchable ...
A {**polymorphic**} language being developed by Stephen Crawley <sxc@itd.dtso.oz.au> of Defence ... **Assembler**/linker, **disassembler**, documentation, examples. ...

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 98 of 302

www.linuxguruz.com/foldoc/Dictionary - Similar pages - Note this

no title
this loses all the advantages of being **stateless** requires patches in the server ... search [Uo] 96-7p6 W3C attempts at organising **metadata** exa WebCrawler, ...
neil.franklin.ch/Underst_Computers - 986k - Cached - Similar pages - Note this

words by the number of messages they appear
listDirectories directoryLister disapprove **disassembler** 3 disasters disclaimer ... specially spiffy squoosh starters stated 14 **stateless** statue stayed stem ...
twistedmatrix.com/users/acapnotic/wip/wordcount-3plus.txt - 284k -
Cached - Similar pages - Note this

[PDF] .NET Tutorialfor Beginners
File Format: PDF/Adobe Acrobat
The **assembly** manifest contains the **assembly's metadata** and it also ... To visualize the IL code Microsoft provides a **disassembler** tool ...
download.microsoft.com/.../e/7/8e725d96-7ec3-498b-9fa7-86779aed101f/dotNET%20Tutorial%20for%20Beginners.pdf - Similar pages - Note this

[Linux Mall] Sempre conectada com você!- [Translate this page]
Obtaining Database **Metadata** Chapter 8. **Tables** and Indexes Section 8.1. ... Platform Specific **Model** - PSM 2. **Model Transformation** 3. Padrões MDA 3. ...
www.linuxmall.com.br/index.php?page=categ/flypage_info&print=1&product_id= -
Similar pages - Note this

[PDF] Hibernate in Action
File Format: PDF/Adobe Acrobat
discuss Hibernate's fully **polymorphic** behavior as well. ... A view may be used to **transform** the **table**-per-subclass **model** into the ...
www.boombala.com/Hibernate.In.Action.pdf - Similar pages - Note this

[PDF] Hibernate in Action
File Format: PDF/Adobe Acrobat
How is mapping **metadata** defined? Since the object/relational transforma- ... A view may be used to **transform** the **table**-per-subclass **model** into the ...
dev2dev.bea.com.cn/bbs/servlet/D2DServlet/download/126-34474-202400-2852/HibernateInAction.pdf - Similar pages - Note this

Englesko - Bosanski Rjecnik: Sve rijeci
Assembly of Prismatic Huul Blocks ... city **transformation models** · city's · CIV Convention ... cryptographic **security** mechanism cryptography · cryptology ...
www.igman.com/rjecnik/bosansko-engleski/rijeci.htm - Similar pages - Note this

Lecture Notes in Computer Science
251 Jolita Ralyté and Colette Rolland An **Assembly** Process **Model** for Method ... and M. von Löwis and R. Schröder Using SDL in a **Stateless** Environment . ...
www.math.utah.edu/pub/tex/bib/toc/lncs2001b.html - 981k -
Cached - Similar pages - Note this

JOS Statistics - Recent Comments
USAID's Food Aid program addresses food **security** issues by: • Developing and implementing **models** that better target food distribution to vulnerable groups, ...
www.usabilitymustdie.com/jos/WW_Weekly_Comments_09_May_2004.html - 907k -
Cached - Similar pages - Note this

JOS Statistics - Recent Comments
Fri 16 Apr | Nam | Regarding compiling the **metadata**; how do you take care ... Does the

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 99 of 302

ShareWare **model** is still viable in 2004 | Sun 11 Apr | Linus Ericsson ...
www.usabilitymustdie.com/jos/WW_Weekly_Comments_18_Apr_2004.html - 951k - Cached - Similar pages - Note this

[PDF] Seminar System Modeling
File Format: PDF/Adobe Acrobat
provides a **model** for the **assembly** of business. solutions from collections of individual ...
So in our example, there is a **stateless** session bean for the ...
wendtstud1.hpi.uni-potsdam.de/sysmod-seminar/SS2006/SysmodSeminar2006.pdf - Similar pages - Note this

Statistics per Wikibook
T TCM / TI / TI 83 Plus **Assembly** / Tablature / **Table** Tennis / Tagalog / Tamil / Taxation ...
U UK Constitution and Government / UNIX Computing **Security** / US ...
en.wikipedia.org/wikistats/wikibooks/EN/Wikibooks_EN.htm - Similar pages - Note this

View Resumes
Designed and wrote EJB 2.0 (**Stateless** Session Bean) for data-access and business ...
Symbol **Table** program for a compiler. 5. **Security** Issues report. ...
www.teleworkrecruiting.com/resume-detail.php?category=IT+and+Computer+Programming - Similar pages - Note this

[PDF] Wielka encyklopedia komputerów
File Format: PDF/Adobe Acrobat
cross **assembler**, 106. cross compiler, 106. cross media, 106 ... obsługują te **model** „wiele do wielu". Obiektowe bazy danych są ...
czytelnia.reporter.pl/rozdzialy/wencko.pdf - Similar pages - Note this

[PDF] Oracle® Database
File Format: PDF/Adobe Acrobat
Java has inheritance, **polymorphism**, and component **models** for developing ... need to store the data in **tables** between each **transformation** stage. ...
www.rdbprime.com/Oracle/Oracle_Docs/Oracle10gDB_Server/appdev.101/b10795.pdf - Similar pages - Note this

[PDF] Oracle® Database
File Format: PDF/Adobe Acrobat
Java has inheritance, **polymorphism**, and component **models** for developing ... **tables** between each **transformation** stage. These functions can accept and return ...
www.sme-solution.com/Oracle/B19306_01/appdev.102/b14251.pdf - Similar pages - Note this

Computer Sweden - Sveriges IT-tidning - dagliga nyheter om it ...
disassembler - backassemblerare eller bakåtassemblerare - program som översätter maskinkod till **assembler**. (Jämför med bakåtkompilering.) ...
cstjanster.idg.se/sprakwebben/ord.asp?ord=som - Similar pages - Note this

3k Delphi Tips
Consuming C# Web Services with Delphi 7 Professional ... .NET **Assembly** Import and Export Wizards for Delphi ... **Stateless** incremental fetching in MIDAS ...
xdds.nepro.net/~cho/3kdt/index2.php?viewall=1&bgcolor= - Similar pages - Note this

the 6187925 of 2941789 and 2682874 to 2560344
... boat 5405 uses 5404 younger 5403 payment 5402 **assembly** 5402 lunch 5400 sought ...
41 subclasses 41 sturgeon 41 statutorily 41 **stateless** 41 stablemate 41 ...
www.cse.yorku.ca/~mack/1020/wordfreq2.txt - 865k - Cached - Similar pages - Note this

dict200k txt

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 100 of 302

... **assembler Assembler** assemblies Assemblies assembling Assembling **assembly** ...
Disarticulation Disarticulator **disassemble disassembler** disassembly ...
www.cs.bsu.edu/gwen/peb/cs232/Projects/SpellCheck/dict200k.txt -
Similar pages - Note this

1 17824229 the 2 11269215 of 3 9836301
File Format: Unrecognized
... ARMS 1512 16590 CONTEMPORARY 1513 16585 **ASSEMBLY** 1514 16567 PREMIER 1515 ... 140 SUDOKU 38617 140 STOREHOUSE 38618 140 **STATELESS** 38619 140 SPROUTS 38620 ...
oc-co.org/files/rank_occs_word.gz - Similar pages - Note this

hell-o visitor on the playerclan site
... states stateside statesman. **stateless** statelessness statelier stateliest ... polynya polyp polyparium polypary. **polymorphic polymorphism** polymorphous ...
www.geocities.com/playerclansite/ - Similar pages - Note this

aux ciar org ttk lemma ttkciar 01 txt
... stars 1294 tungsten 1294 **table** 1294 laser 1294 directory 1294 decoration ... 51 bhorsepic 51 beats 51 atsushi 51 **assembler** 51 arsenic 51 anita 51 alumni ...
aux.ciar.org/ttk/lemma.ttkciar.01.txt - Similar pages - Note this

Search Engine Page
pa · PABX · PACE · Packard Bell Electronics, Inc. · packed decimal · Packed Encoding Rules · packet · packet-switched · Packet **Assembler/Disassembler** ...
thornesz.tripod.com/id17.html - 977k - Cached - Similar pages - Note this

1017 Internet 961 protocol 932 Unix 928 TCP/IP 911 search engine ...
... 15 Parallel Sysplex 15 Packet **Assembler/Disassembler** 15 PRECCX 15 PD 15 PCI ... 14 affine **transformation** 14 administrator 14 ad-hoc **polymorphism** 14 a+ ...
www.kambarys.lt/test.php - 854k - Cached - Similar pages - Note this

: W : GTI - Glosario Terminología Informática- [Translate this page]
Glosario de terminología informática Ingles-Español. Términos que se emplean en el campo de la Informática y otras ciencias que le son afines.
www.tugurium.com/gti/termino.asp?tr=W - 6k - Cached - Similar pages - Note this

00databasealphabet I+nX u 00databaseinfo B0 EQ 00databasescript I+ ...
... qZn z **assembler** qaa u assembling qbl Bk **assembly** qcs Bo **assembly** average ... disarrange CPSA m disarray CPSm BP **disassemble** CPT1 x **disassembler** CPUm o ...
fido.sv.mh.ru/~stas/dict/sokrat_en-ru.index - 970k - Cached - Similar pages - Note this

軍事用語：一般用語- [Translate this page]
Air Force **Director** of Intelligence, Surveillance, and Reconnaissance ... Air Force National **Security** and Emergency Preparedness Agency ...
www.komodic.flnet.org/komo-dic/terms/komo-terms.html - 1k -
Cached - Similar pages - Note this

In order to show you the most relevant results, we have omitted some entries very similar to the 46 already displayed.
If you like, you can repeat the search with the omitted results included.

Download Google Pack: free essential software for your PC

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 101 of 302

RECEIVED
CENTRAL FAX CENTER
MAR 13 2007

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200400406-1

Inventor(s): Steven R. Carr et al.

Application No.: 10/868,434

Filing Date: June 14, 2004

Confirmation No.: 4452

Examiner: Lu, Kuen S

Group Art Unit: 2167

Title: DATA SERVICES HANDLER

Mail Stop Amendment
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL LETTER FOR RESPONSE/AMENDMENT

Transmitted herewith is/are the following in the above-identified application:

- [x] Response/Amendment
- [ ] New fee as calculated below
- [ ] No additional fee
- [ ] Other
- [ ] Petition to extend time to respond
- [ ] Supplemental Declaration

Fee$

CLAIMS AS AMENDED BY OTHER THAN A SMALL ENTITY						
(1) FOR	(2) CLAIMS REMAINING AFTER AMENDMENT	(3) NUMBER EXTRA	(4) HIGHEST NUMBER PREVIOUSLY PAID FOR	(5) PRESENT EXTRA	(6) RATE	(7) ADDITIONAL FEES
TOTAL CLAIMS	36	MINUS	39	= 0	X $50	$ 0
INDEP. CLAIMS	5	MINUS	8	= 0	X $200	$ 0
☐ FIRST PRESENTATION OF A MULTIPLE DEPENDENT CLAIM					+ $360	$ 0
EXTENSION FEE	☐ 1st Month $120	☐ 2nd Month $450	☐ 3rdMonth $1020	☐ 4th Month $1590		$ 0
					OTHER FEES	$
					TOTAL ADDITIONAL FEE FOR THIS AMENDMENT	$ 0

Charge $ 0 to Deposit Account 08-2025. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number (571) 273-8300.
Date of facsimile: March 13, 2007

Typed Name: Joy C. Ngo

Signature:

Respectfully submitted,

Steven R. Carr et al.

By

Ken J. Koestner
Attorney/Agent for Applicant(s)

Reg No. : 33,004

Date : March 13, 2007

Telephone : (949) 251-0250

Rev 10/05 (TransAmdFax)

RECEIVED
CENTRAL FAX CENTER

MAR 1 3 2007

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200400406-1

Inventor(s): Steven R. Carr et al.

Application No.: 10/868,434

Filing Date: June 14, 2004

Confirmation No.: 4452

Examiner: Lu, Kuen S

Group Art Unit: 2167

Title: DATA SERVICES HANDLER

Mail Stop Amendment
Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL LETTER FOR RESPONSE/AMENDMENT

Transmitted herewith is/are the following in the above-identified application:

- [x] Response/Amendment
- [ ] New fee as calculated below
- [ ] No additional fee
- [ ] Other
- [ ] Petition to extend time to respond
- [ ] Supplemental Declaration

Fee$

CLAIMS AS AMENDED BY OTHER THAN A SMALL ENTITY						
(1) FOR	(2) CLAIMS REMAINING AFTER AMENDMENT	(3) NUMBER EXTRA	(4) HIGHEST NUMBER PREVIOUSLY PAID FOR	(5) PRESENT EXTRA	(6) RATE	(7) ADDITIONAL FEES
TOTAL CLAIMS	36	MINUS	39	= 0	X $50	$ 0
INDEP. CLAIMS	5	MINUS	8	= 0	X $200	$ 0
☐ FIRST PRESENTATION OF A MULTIPLE DEPENDENT CLAIM					+ $360	$ 0
EXTENSION FEE	☐ 1st Month $120	☐ 2nd Month $450	☐ 3rdMonth $1020	☐ 4th Month $1590		$ 0
					OTHER FEES	$
					TOTAL ADDITIONAL FEE FOR THIS AMENDMENT	$ 0

Charge $ 0 to Deposit Account 08-2025. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number (571) 273-8300.
Date of facsimile: March 13, 2007

Typed Name: Joy C. Ngo
Signature:

Respectfully submitted,
Steven R. Carr et al.
By
Ken J. Koestner
Attorney/Agent for Applicant(s)
Reg No. : 33,004
Date : March 13, 2007
Telephone : (949) 251-0250

Rev 10/00 (TransAmdFax)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Steven R. Carr, Venkatakrishnan Muthuswamy, and Tudor Har		
Assignee:	Hewlett-Packard Development Company, L.P.		
Title:	DATA SERVICES HANDLER		
Serial No.:	10/868,434	Filing Date:	June 14, 2004
Examiner:	Lu, Kuen S	Group Art Unit:	2167
Docket No.:	200400406-1	Confirmation No.:	4452

Irvine, California
March 13, 2007

MAIL STOP AMENDMENT
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE TO OFFICE ACTION DATED DECEMBER 13, 2006

Dear Sir:

This paper is responsive to the Office Action dated December 13, 2006, having a shortened statutory period expiring March 13, 2007. Further examination and reconsideration are respectfully requested.

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

Remarks/Arguments begin on page 12 of this paper.

KOESTNER BERTANI LLP
18662 MACARTHUR BLVD.
SUITE 400
IRVINE, CA 92612
TEL (949) 251-0250
FAX (949) 251-0260

PAGE 3/15 * RCVD AT 3/13/2007 8:16:28 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/17 * DNIS:2738300 * CSID:9492510260 * DURATION (mm-ss):04-22

RECEIVED
CENTRAL FAX CENTER

MAR 13 2007

AMENDMENTS TO THE CLAIMS

The claims are as follows:

1. (Currently Amended): A data services handler comprising:
an interface between a data store and applications that supply and consume data; and
a real time information director (RTID) that transforms data for supply and/or consumption by the applications under direction of polymorphic metadata that defines a security model and data integrity rules for application to the data, the RTID configured to dynamically instantiate, create, and cache for reuse disassemblers and assemblers according to demand, the RTID further comprising:
at least one disassembler defined according to the polymorphic metadata that disassembles inbound data documents into multiple records in preparation for insertion into metadata tables; and
at least one assembler defined according to the polymorphic metadata that assembles outbound data documents from at least one record selected from at least one metadata-selected table.

2. (Original): The data services handler according to Claim 1 further comprising:
a data mapper capable of collecting data from a plurality of diverse applications that can supply data in multiple diverse external document formats and mapping the supplied data to a predetermined unified data model according to the polymorphic metadata.

3. (Original): The data services handler according to Claim 1 wherein:
the real time information director comprises a logic that cleanses and enriches the data, supplies data security, and insures data integrity.

KOESTNER BERTANI LLP
15663 MACARTHUR BLVD.
SUITE 400
IRVINE, CA 92612
TEL (949) 251-0250
FAX (949) 251-0260

PAGE 4/15 * RCVD AT 3/13/2007 8:16:28 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/17 * DNIS:2738300 * CSID:9492510260 * DURATION (mm-ss):04-22

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 105 of 302

4. (Currently Amended): The data services handler according to Claim 1 wherein the real time information director further comprises:

~~a disassembler that disassembles inbound data documents into multiple records in preparation for insertion into metadata tables; and~~

~~an assembler that assembles outbound data documents from at least one record selected from at least one metadata selected table~~

the RTID configured to use a same descriptive polymorphic metadata to build both assemblers and disassemblers for a same document class.

5. (Original): The data services handler according to Claim 1 wherein the real time information director further comprises:

a logic that defines a data model according to received descriptive metadata that identifies individual data elements and specifies logic behaviors applicable to the individual data elements.

6. (Original): The data services handler according to Claim 1 wherein the real time information director further comprises:

a logic that represents metadata as polymorphic objects, enabling individual metadata objects to have distinct numbers and types of attributes according to a predetermined object class, the behaviors and attributes being defined specific to object class.

7. (Original): The data services handler according to Claim 1 wherein the real time information director further comprises:

a logic that represents metadata as stateless polymorphic objects that are permanent after initial construction, whereby methods of stateful metadata classes delegate operation to the stateless polymorphic metadata objects.

KOESTNER BERTANI LLP
18662 MACARTHUR BLVD.
SUITE 400
IRVINE, CA 92612
TEL: (949) 251-0250
FAX: (949) 251-0260

8. (Original): The data services handler according to Claim 1 wherein the real time information director further comprises:

PAGE 5/15 * RCVD AT 3/13/2007 8:16:28 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/17 * DNIS:2738300 * CSID:9492510260 * DURATION (mm-ss):04-22

at least one metadata object constructed using constructors of multiple metadata classes, the constructors having parameters that specify metadata data elements and being nested to represent hierarchical relationships between the specified metadata data elements, the at least one metadata object including identifying information for deriving the hierarchical relationships.

9. (Currently Amended): The data services handler according to Claim 1 wherein the real time information director further comprises:

a logic that uses received data document mapping metadata to disassemble elements of inbound data documents into rows and columns to be inserted into inbound metadata-selected tables in a database, select rows and columns from outbound metadata-selected tables of the database, ~~and~~ assemble elements from the outbound metadata-selected tables into an outbound data document, and generate a schema for documents that can be input or output using the received data document mapping metadata.

10. (Original): The data services handler according to Claim 1 wherein:

the real time information director receives and publishes data documents having a content and a format defined by document metadata.

11. (Original): The data services handler according to Claim 1 wherein the real time information director further comprises:

a logic that reads a schema of tables identified in metadata; and

a logic that automatically generates and caches prepare SQL statements from a content metadata portion of document metadata according to the schema.

KOESTNER BERTANI LLP
18662 MACARTHUR BLVD
SUITE 400
IRVINE, CA 92612
TEL (949) 251-0250
FAX (949) 251-0260

PAGE 6/15 * RCVD AT 3/13/2007 8:16:28 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/17 * DNIS:2738300 * CSID:9492510260 * DURATION (mm-ss):04-22

12. (Original): The data services handler according to Claim 1 further comprising:

a servlet that invokes the real time information director synchronously when an inbound data document is delivered via HyperText Transport Protocol (HTTP);

a logic that invokes the real time information director asynchronously when an inbound data document is delivered via Java Message Service (JMS);

a servlet that invokes the real time information director synchronously when an outbound data document is requested via HyperText Transport Protocol (HTTP); and

a logic that invokes the real time information director asynchronously when an outbound data document is requested via Java Message Service (JMS).

13. (Original): The data services handler according to Claim 1 wherein the real time information director further comprises:

a subscription handler that responds to a triggering event by publishing a document according to content and format specified by Document Metadata, and sending the document to a designated queue or topic.

14. (Original): The data services handler according to Claim 1 wherein the real time information director further comprises:

a security manager that identifies an agent on behalf of whom information is queried or updated in a document request, dynamically assigns an internal unique identifier to consumers and agents upon registration; and

the real time information director grants access to individual documents based on security policies according to a relationship between an agent, a consumer affected by the agent, and permissions granted by the consumer.

KOESTNER BERTANI LLP
18002 MACARTHUR BLVD.
SUITE 400
IRVINE CA 92612
TEL (949) 251-0250
FAX (949) 251-0260

PAGE 7/15 * RCVD AT 3/13/2007 8:16:28 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/17 * DNIS:2738300 * CSID:9492510260 * DURATION (mm-ss):04-22

15. (Original): The data services handler according to Claim 14 wherein: the security manager enforces securities policies that are specifiable for each document.

16. (Original): The data services handler according to Claim 14 wherein: the security manager enforces securities policies that are customized to identify specific agent and consumer relationships expressed in a business model.

17. (Original): The data services handler according to Claim 1 wherein the real time information director further comprises:
an audit log handler that responds to changes in the data store by compiling and storing information based on metadata that determines content and detail of the compiled information.

18. (Original): The data services handler according to Claim 1 wherein the real time information director further comprises:
a data cleanser that receives data elements identified by Document Metadata, normalizes the data elements, queues source and destination information for detection of duplicates, and cross-references duplicates in the data store.

19. (Original): The data services handler according to Claim 1 wherein the real time information director further comprises:
an aggregation engine that computes aggregates in memory from data selected from the data store.

20. (Original): The data services handler according to Claim 1 wherein the real time information director further comprises:
a real-time scoring engine.

KOESTNER BERTANI LLP
18662 MACARTHUR BLVD
SUITE 400
IRVINE, CA 92612
TEL (949) 251-0250
FAX (949) 251-0260

PAGE 8/15 * RCVD AT 3/13/2007 8:16:28 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/17 * DNIS:2738300 * CSID:9492510260 * DURATION (mm-ss):04-22

21. (Original): The data services handler according to Claim 1 wherein the real time information director further comprises:

a business rules engine that defines a business object model capable of enabling access to data described by Document Metadata, performing computations on data relevant to business rules specified by the business object model, concluding a decision based on computation results, and executing the decision.

22. (Currently Amended): An article of manufacture comprising:

a controller-usable storage medium having a computable readable program code embodied therein for managing data services, the computable readable program code further comprising:

a code capable of causing the controller to transfer data among a data store and applications that supply and consume data; ~~and~~

a code capable of causing the controller to transform data for transfer using a representation of metadata as polymorphic objects, enabling individual metadata objects to have distinct numbers and types of attributes according to a predetermined object class, the behaviors and attributes being defined specific to object class;

a code capable of causing the controller to dynamically instantiate, create, and cache for reuse at least one disassembler and at least one assembler according to demand,

a code dynamically created by the controller as defined by polymorphic metadata that disassembles inbound data documents into multiple records in preparation for insertion into metadata tables; and

a code dynamically created by the controller as defined by polymorphic metadata that assembles outbound data documents from at least one record selected from at least one metadata-selected table.

KOESTNER BERTANI LLP
18662 MACARTHUR BLVD.
SUITE 400
IRVINE, CA 92612
TEL (949) 251-0250
FAX (949) 251-0260

PAGE 9/15 * RCVD AT 3/13/2007 8:16:28 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/17 * DNIS:2738300 * CSID:9492510260 * DURATION (mm-ss):04-22

23. (Original): The article of manufacture according to Claim 22 further comprising:

a code capable of causing the controller to transform data under direction of polymorphic metadata that defines a security model and data integrity rules for application to the data.

24. (Original): The article of manufacture according to Claim 22 further comprising:

a code capable of causing the controller to collect data from a plurality of diverse applications that can supply data in multiple diverse external document formats and mapping the supplied data to a predetermined unified data model according to the polymorphic metadata.

25. (Original): The article of manufacture according to Claim 22 further comprising:

a code capable of causing the controller to represent metadata as stateless polymorphic objects that are permanent after initial construction, whereby methods of stateful metadata classes delegate operation to the stateless polymorphic metadata objects.

26. (Original): The article of manufacture according to Claim 22 further comprising:

a code capable of causing the controller to construct at least one metadata object using constructors of multiple metadata classes;

a code capable of causing the controller to use constructor parameters to specify metadata data elements and being nested to represent hierarchical relationships between the specified metadata data elements, the at least one metadata object including identifying information for deriving the hierarchical relationships.

KOESTNER BERTANI LLP
18662 MacARTHUR BLVD.
SUITE 400
IRVINE, CA 92612
TEL (949) 251-0250
FAX (949) 251-0260

PAGE 10/15 * RCVD AT 3/13/2007 8:16:28 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/17 * DNIS:2738300 * CSID:9492510260 * DURATION (mm-ss):04-22

27. (Original): The article of manufacture according to Claim 22 further comprising:

a code capable of causing the controller to use received data document mapping metadata to disassemble elements of inbound data documents into rows and columns to be inserted into inbound metadata-selected tables in a database, select rows and columns from outbound metadata-selected tables of the database and assemble elements from the outbound metadata-selected tables into an outbound data document, and generate a schema for documents that can be input or output using the received data document mapping metadata.

28. (Currently Amended): An enterprise system comprising:

a real time information director (RTID) that manages information flow between a data store and applications that supply and consume data~~, including transforming data under direction of~~ and dynamically instantiates, creates, and caches for reuse disassemblers and assemblers according to demand, the RTID configured to create at least one disassembler defined according to the polymorphic metadata that disassembles inbound data documents into multiple records in preparation for insertion into metadata tables and create at least one assembler defined according to the polymorphic metadata that assembles outbound data documents from at least one record selected from at least one metadata-selected table.

29. (Original): The enterprise system according to Claim 28 further comprising:

a business rules engine coupled to the real time information director that personalizes transactions by applying business rules and prediction models against cached entity data.

KOESTNER BERTANI LLP
18662 MACARTHUR BLVD.
SUITE 400
IRVINE, CA 92612
TEL (949) 331-0230
FAX (949) 331-0260

PAGE 11/15 * RCVD AT 3/13/2007 8:16:28 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/17 * DNIS:2738300 * CSID:9492510260 * DURATION (mm-ss):04-22

30. (Original): The enterprise system according to Claim 28 further comprising:

a server platform coupled to the real time information director that serves deployed models, customer data, and interaction data.

31. (Original): The enterprise system according to Claim 28 wherein:

the RTID represents metadata as polymorphic objects, enabling individual metadata objects to have distinct numbers and types of attributes according to a predetermined object class, the behaviors and attributes being defined specific to object class.

32. (Original): The enterprise system according to Claim 28 wherein:

the RTID represents metadata as stateless polymorphic objects that are permanent after initial construction, whereby methods of stateful metadata classes delegate operation to the stateless polymorphic metadata objects.

33. (Currently Amended): A data services handler comprising:

a real time information director (RTID) that maps documents into a data model further comprising:

a disassembly logic that is dynamically created by the RTID as defined by polymorphic metadata and disassembles documents for loading into a database;

an assembly logic that is dynamically created by the RTID as defined by polymorphic metadata and assembles documents from records in the database; and

a generator logic that is dynamically created by the RTID as defined by polymorphic metadata and generates an eXtended Markup Language (XML) Schema that describes format of the documents,

the disassembly, assembly, and generator logic being operative using a single descriptive polymorphic metadata.

KOESTNER BERTANI LLP
18662 MACARTHUR BLVD.
SUITE 400
IRVINE, CA 92612
TEL (949) 251-0252
FAX (949) 251-0260

PAGE 12/15 * RCVD AT 3/13/2007 8:16:28 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/17 * DNIS:2738300 * CSID:9492510260 * DURATION (mm-ss):04-22

34. (Original): The data services handler according to Claim 33 wherein the metadata further comprises:

content metadata that defines data to be included into a document; and

format metadata that defines how the data appears in the document.

35. (Original): The data services handler according to Claim 33 wherein the real time information director further comprises:

a query logic that queries the database to determine information about tables in the database that the document uses, the information including field name, field datatype, and field allowance of nulls.

36. (Currently Amended): A method for generating native documents as an eXtended Markup Language (XML) representation of data in a Structured Query Language (SQL) database, the method comprising:

dynamically creating metadata for a new native document as defined by polymorphic metadata further comprising:

determining whether inbound documents are inserted, delserted or upserted;

generating native documents;

determining what data from the SQL database is sufficient for usage in the new native document; and

creating an XML Schema for the native document.

37-39. (Canceled)

KOESTNER BERTANI LLP
18662 MACARTHUR BLVD.
SUITE 400
IRVINE, CA 92612
TEL (949) 251-0250
FAX (949) 251-0260

PAGE 13/15 * RCVD AT 3/13/2007 8:16:28 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/17 * DNIS:2738300 * CSID:9492510260 * DURATION (mm-ss):04-22

RECEIVED
CENTRAL FAX CENTER
MAR 1 3 2007

REMARKS

The above-identified application is United States application serial number 10/868,434 filed on June 14, 2004. Claims 1-39 are pending in the application. Claims 1-39 are rejected.

Claim Objections

Claims 9 and 37 are objected because of informalities. Applicants have amended Claim 9 to clarify wording as noted by the Examiner. Applicants have canceled Claim 37 without prejudice.

Rejection of Claim Under 35 USC §112

Claims 22-27 and 38-39 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. Applicants traverse the rejection of Claims 22-27 because "controller-usable medium" is fully described in the specification in paragraph [0036]. Applicants have canceled Claims 38-39.

Claim 36 is rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential steps, such omission amounting to a gap between the steps. Applicants have amended Claim 36 as suggested.

Rejection of Claim Under 35 USC §101

Claims 1-39 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. Applicants traverse the rejection of Claims 1-21 and 28-36 because the specification and drawings clearly show that the "data services handler" and the "enterprise system" are machines including a hardware and software combination.

KOESTNER BERTANI LLP
18662 MACARTHUR BLVD
SUITE 400
IRVINE, CA 92612
TEL (949) 251-0250
FAX (949) 251-0260

Applicants have amended Claims 22-27 to specify a "storage medium" so that the product is within a category of patentable storage matter.

PAGE 14/15 * RCVD AT 3/13/2007 8:16:28 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/17 * DNIS:2738300 * CSID:9492510260 * DURATION (mm-ss):04-22

RECEIVED
CENTRAL FAX CENTER

MAR 13 2007

Rejection of Claim Under 35 USC §103

Claims 1-39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brendle et al. (US Patent Application 2005/0021354, hereafter "Brendle") in view of Reed et al. (US Patent 6,088,717), hereafter "Reed"). Applicants have amended the Claims. The amended Claims 1-36 distinguish over Brendle in view of Reed at least because the references fail to disclose "a real time information director (RTID) . . .configured to dynamically instantiate, create, and cache for reuse disassemblers and assemblers according to demand" or disassemblers and assemblers that are dynamically created as defined by polymorphic metadata.

Applicants have canceled Claims 37-39.

CONCLUSION

Applicants believe the application, including Claims 1-36 are in form for allowance and a notice to that effect is solicited. In the event it would facilitate prosecution of this application, the Examiner is invited to telephone the undersigned at (949) 251-0250.

I hereby certify that this correspondence is being facsimile transmitted to the USPTO, Central Number at (571) 273-8300 on the date shown below:

(Signature)

Joy C. Ngo
(Printed Name of Person Signing Certificate)

March 13, 2007
(Date)

Respectfully submitted,

Ken J. Koestner
Attorney for Applicant(s)
Reg. No. 33,004

KOESTNER BERTANI LLP
18662 MACARTHUR BLVD
SUITE 400
IRVINE, CA 92612
TEL (949) 251-0250
FAX (949) 251-0260

PAGE 15/15 * RCVD AT 3/13/2007 8:16:28 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/17 * DNIS:2738300 * CSID:9492510260 * DURATION (mm-ss):04-22

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 10/868,434	Filing Date 06/14/2004	☐ To be Mailed

APPLICATION AS FILED – PART I

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

APPLICATION AS AMENDED – PART II

AMENDMENT 03/13/2007	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 36	Minus	** 39	= 0	X $ =		OR	X $50=	0
Independent (37 CFR 1.16(h))	* 5	Minus	***8	= 0	X $ =		OR	X $200=	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:
Rosalind Ball

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.
10/868,434	06/14/2004	Steven R. Carr	200400406-1	4452

22879 7590 12/13/2006

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
LU, KUEN S	

ART UNIT	PAPER NUMBER
2167	

DATE MAILED: 12/13/2006

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

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)	
	10/868,434	CARR ET AL.	
	Examiner	Art Unit	
	Kuen S. Lu	2167	

-- 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 *14 June 2006*.

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

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

Disposition of Claims

4)☒ Claim(s) *1-39* 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 *14 June 2004* 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 *6/14/2004*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. The Action is responsive to Applicant's Application filed June 14, 2004. Please note Claims 1 and 2-39 are pending.

Information Disclosure Statement

2. The information disclosure statement filed June 14, 2004 is in compliance with 37 CFR 1.97(c) and therein has been considered. Its corresponding PTO-1449 has been electronically signed as attached.

Drawings

3. The drawings, filed June 14, 2004, are considered in compliance with 37 CFR 1.81 and accepted.

Claim Objections

4. Claims 9 and 37 are objected to because of the following informalities:

As for claim 9, the claim recites a logic that uses data to disassemble elements, select rows and columns, **and** assemble elements, **and** generate a schema. The claim seems to recite a logic that uses data to disassemble elements, selects rows and columns, assembles elements **and** generates a schema? Or the claim seems to recite a logic that uses data to: disassemble elements, select rows and columns, assemble elements **and** generate a schema? Appropriate correction is required.

As for claim 37, the element of SQL connections and prepared statements includes "document handlers" twice. The claim seems to recite a component repeatedly in a same element by requirement or by accident? Appropriate correction is required.

Claim Rejections - 35 USC § 112

5.1. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

5.1.1. Claims 22-27 and 38-39 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

As for claims 22-27 and 38, the claims describe "controller-usable medium" which is not described in the specification.

As for claim 39, the claim describes five "means for"s performing some functional steps, however, the "means for"s are not described in the specification.

5.2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5.2.1. Claim 36 is rejected under 35 U.S.C. 112, second paragraph, as being incomplete for omitting essential steps, such omission amounting to a gap between the steps. See MPEP § 2172.01.

As per claims 36, there exists at least one omitted step: generating native documents.

Claim Rejections - 35 USC § 101

6.1. 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.

6.1.1. As set forth in MPEP 2106 (II) (A):

The claimed invention as a whole must accomplish a practical application. That is, it must produce a "useful, concrete and tangible result." State Street, 149 F.3d at 1373, 47 USPQ2d at 1601-02. The purpose of this requirement is to limit patent protection to inventions that possess a certain level of "real world" value, as opposed to subject matter that represents nothing more than an idea or concept, or is simply a starting point for future investigation or research (Brenner v. Manson, 383 U.S. 519, 528-36, 148 USPQ 689, 693-96); In re Ziegler, 992, F.2d 1197, 1200-03, 26 USPQ2d 1600, 1603-06 (Fed. Cir. 1993)). Accordingly, a complete disclosure should contain some indication of the practical application for the claimed invention, i.e., why the applicant believes the claimed invention is useful.

Apart from the utility requirement of 35 U.S.C. 101, usefulness under the patent eligibility standard requires significant functionality to be present to satisfy the useful result aspect of the practical application requirement. See Arrhythmia, 958 F.2d at 1057, 22 USPQ2d at 1036. Merely claiming nonfunctional descriptive material stored in a computer-readable medium does not make the invention eligible for patenting. For example, a claim directed to a word processing file stored on a disk may satisfy the utility requirement of 35 U.S.C. 101 since the information stored may have some "real world" value. However, the mere fact that the claim may satisfy the utility requirement of 35 U.S.C. 101 does not mean that a useful result is achieved under the practical application requirement. The claimed invention as a whole must produce a "useful, concrete and tangible" result to have a practical application

6.1.2. As set forth in MPEP 2106 (IV) (B) (1):

Claims to computer-related inventions that are clearly nonstatutory fall into the same general categories as nonstatutory claims in other arts, namely natural phenomena such as magnetism, and abstract ideas or laws of nature which constitute "descriptive material." Abstract ideas, Warmerdam, 33 F.3d at 1360, 31 USPQ2d at 1759, or the mere manipulation of abstract ideas, Schrader, 22 F.3d at 292-93, 30 USPQ2d at 1457-58, are not patentable. Descriptive material can be characterized as either "functional descriptive material" or "nonfunctional descriptive material." In this context, "functional descriptive material" consists of data structures and computer programs which impart functionality when employed as a computer component. (The definition of "data structure" is "a physical or logical relationship among data elements, designed to support specific data manipulation functions." The New IEEE Standard Dictionary of Electrical and Electronics Terms 308 (5th ed. 1993).) "Nonfunctional descriptive material" includes but is not limited to music, literary works and a compilation or mere arrangement of data. Both types of "descriptive material" are nonstatutory when claimed as descriptive material per se. Warmerdam, 33 F.3d at 1360, 31 USPQ2d at 1759. When functional descriptive material is recorded on some computer-readable medium it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized. Compare In re Lowry, 32 F.3d 1579, 1583-84, 32 USPQ2d 1031, 1035 (Fed. Cir. 1994) (claim to data

structure stored on a computer readable medium that increases computer efficiency held statutory) and Warmerdam, 33 F.3d at 1360-61, 31 USPQ2d at 1759 (claim to computer having a specific data structure stored in memory held statutory product-by-process claim) with Warmerdam, 33 F.3d at 1361, 31 USPQ2d at 1760 (claim to a data structure per se held nonstatutory).

6.1.3. As set forth in MPEP 2106 (IV)(B)(1)(a):

Similarly, computer programs claimed as computer listings *per se, i.e.,* the descriptions or expressions of the programs, are not physical "things." They are neither computer components nor statutory processes, as they are not "acts" being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer which permit the computer program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element which defines structural and functional interrelationships between the computer program and the rest of the computer which permit the computer program's functionality to be realized, and is thus statutory. Accordingly, it is important to distinguish claims that define descriptive material *per se from claims* that define statutory inventions.

Products may be either machines, manufactures, or compositions of matter.
A machine is "a concrete thing, consisting of parts or of certain devices and combinations of devices. *Burr v. Duryee. 68* U.S. (1 Wall.) *531, 570 (1863).*
If a claim defines a useful machine or manufacture by identifying the physical structure of the machine or manufacture in terms of its hardware or hardware and software combination, it defines a statutory product. See, e.g., *Lowry, 32 F.3d* at *1583, 32* USPQ2d at *1034-35; Warmerdarn,* 33 F.3d at 1361-62, 31 USPQ2d at 1760.
Office personnel must treat each claim as a whole. The mere fact that a hardware element is recited in a claim does not necessarily limit the claim to a specific machine or manufacture. Cf. *In re Iwahashi, 888* F.2d 1370, 1374-75, 12 USPQ2d 1908, *191 1*-12 (Fed. Cir. 1989), cited with approval in *Alappat,* 33 F.3d at 1544 n.24, 31 USPQ2d at 1558 n_24.

6.2. Claims 1-39 are rejected under 35 U.S.C. § 101 because the claimed invention is directed to non-statutory subject matter.

As per claim 1, a data services handler comprises an interface that supplies and consumes data, and a real time information director that transforms data. The data services handler seems to be a data structure or software, however, a data structure or software does not belong to any one of statutory categories, and the handler by self does not supply or consume without being in the form of program product, being resident on computer readable storage medium and further being enabled to be executable and executed by a computer processor. Therefore, the data services

handler does not belong to any one of statutory categories and does not ensue any useful result. However, a tangible, concrete and useful result is required in a practical application test. Again, the consequence is non-statutory.

As per claim 22, an article of manufacture comprises controller-usable medium having computer program code embodied for managing data services. However, the claimed "controller-usable" medium comprises wireless telecommunication signals and carrier waves, forms of energy. As forms of energy, the signals and waves are not a matter, composition of matter or product; and do not fall within any one of categories of patentable subject matter. For further rejecting the claim under 35 USC §103 or 35 USC §102, Examiner interprets "machine readable medium" as "computer-readable storage medium".

As per claim 28, the claim is an enterprise system comprising a director that manages information. However, the system does not have specific hardware to support the director to manage information. For example, the director "that manages" by self does not have a specific hardware serving as a means for providing result. Therefore, the claim lacks the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 35 USC 101. They are clearly not a series of steps or acts to be a process nor are they a combination of chemical compounds to be a composition of matter. They are, at best, functional descriptive material per se. As such, they fail to fall within a statutory category.

As per claim 33, a data services handler comprises a real time information director that maps document into a data model. The data services handler seems to be a data structure or software, however, a data structure or software does not belong to any one of statutory categories, and the handler by self does not map without being in the form of program product, being resident on computer readable storage medium and further being enabled to be executable and executed by a computer processor. Therefore, the data services handler does not belong to any one of statutory categories and does not ensue any useful result. However, a tangible, concrete and useful result is required in a practical application test. Again, the consequence is non-statutory.

As per claim 36, the claim directs to a method **generating native documents** comprising **creating metadata for a new native document**. Noted is the step of **creating metadata for a new native document** does not ensue any useful result and does not perform **generating native documents.** However, a tangible, concrete and useful result is required in a practical application test. Again, the consequence is non-statutory.

As per claim 37, the claim is a content management system comprises content data classes, a SQL schema and SQL connections and prepared statements. However, the system does not have specific hardware to support content management. For example, the content management by self does not have a specific hardware serving as a means

for ensuing result. Therefore, the claim lacks the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 35 USC 101. They are clearly not a series of steps or acts to be a process nor are they a combination of chemical compounds to be a composition of matter. They are, at best, functional descriptive material per se. As such, they fail to fall within a statutory category.

As per claim 38, an article of manufacture comprises controller-usable medium having computer program code embodied for generating native documents. However, the claimed "controller-usable" medium comprises wireless telecommunication signals and carrier waves, forms of energy. As forms of energy, the signals and waves are not a matter, composition of matter or product; and do not fall within any one of categories of patentable subject matter. For further rejecting the claim under 35 USC §103 or 35 USC §102, Examiner interprets "machine readable medium" as "computer-readable storage medium".

As per claim 39, the claim is a content management system comprising five "means for" performing various functional steps. However, the system does not have specific hardware or means for supporting the content management. For example, the content management by self does not have a specific hardware or means for generating native documents and ensuing result from the act of generation. Therefore, the claim lacks the necessary physical articles or objects to constitute a machine or a manufacture within the meaning of 35 USC 101. They are clearly not a series of steps or acts to be a

process nor are they a combination of chemical compounds to be a composition of matter. They are, at best, functional descriptive material per se. As such, they fail to fall within a statutory category.

As per claims in the groups (2-21), (23-27), (29-32) and (34-35), the claims inherit the deficiency of being non-statutory directly or indirectly from claims 1, 22, 28 and 33, respectively, and do not remedy the deficiency individually or by inheritance. The consequence is non-statutory.

Claim Rejections - 35 USC § 103

7. 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.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

7.1. Claims 1-39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Brendle et al. (U.S. Patent Application 2005/0021354, hereafter "Brendle") in view of Reed et al. (U.S. Patent 6,088,717, hereafter "Reed").

As per claim 22, Brendle teaches "An article of manufacture comprising: a controller-usable medium having a computable readable program code embodied therein for managing data services, the computable readable program code" (See [0005] and [0049]-[0051] where instructions stored on machine-readable medium for processing application business object and service manager proxy, RPE, RPE proxy and service manager are provided serving as an intermediary between front end application program and service provider) further comprising:

"a code capable of causing the controller to transfer data among a data store and applications that supply and consume data" (See [0049]-[0051] where service manager proxy, RPE, RPE proxy and service manager provide data transfer between database, repository and applications).

Brendle does not explicitly teach "a code capable of causing the controller to transform data for transfer using a representation of metadata as polymorphic objects, enabling individual metadata objects to have distinct numbers and types of attributes according to a predetermined object class, the behaviors and attributes being defined specific to object class".

However, Reed teaches "a code capable of causing the controller to transform data for transfer using a representation of metadata as polymorphic objects, enabling individual metadata objects to have distinct numbers and types of attributes according to a predetermined object class, the behaviors and attributes being defined specific to object class" (See Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 20, lines 55-59 and col. 21, lines 32-36 where exchange of data, metadata, methods

and archiving and retrieval of changed objects can be automatically carried out by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Reed with Brendle reference because both references are directed to providing sets of data elements and associated sets of operations where Reed teaches extending service-oriented business frameworks and processing business objects while Brendle teaches more on communication system for transferring data, metadata and methods from service provider to consumer, and the combined teaching of the references would have enabled Brendle's system to utilize Reed's metadata to define a control structure for communicating front end and back end components easily and quickly because communications as defined could have been automatically established and updated.

As per claim 1, Brendle teaches "A data services handler comprising" (See [0005] and [0049]-[0051] where application business object and service manager proxy, RPE, RPE proxy and service manager are provided serving as an intermediary between front end application program and service provider):
"an interface between a data store and applications that supply and consume data" (See [0049]-[0051] where service manager proxy, RPE, RPE proxy and service

manager are the interface providing data transfer between database, repository and applications).

Brendle does not explicitly teach "a real time information director (RTID) that transforms data under direction of polymorphic metadata that defines a security model and data integrity rules for application to the data".

However, Reed teaches "a real time information director (RTID) that transforms data under direction of polymorphic metadata that defines a security model and data integrity rules for application to the data" (See Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 14, lines 56-60; col. 20, lines 55-59 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically carried out in real-time communication by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Reed with Brendle reference because both references are directed to providing sets of data elements and associated sets of operations where Reed teaches extending service-oriented business frameworks and processing business objects while Brendle teaches more on communication system for transferring data, metadata and methods from service provider to consumer, and the combined teaching of the references would have enabled Brendle's system to utilize Reed's metadata to define a control structure for

communicating front end and back end components easily and quickly because communications as defined could have been automatically established and updated.

As per claim 28, Brendle teaches "An enterprise system" (See [0005] and [0049]-[0051] where an enterprise system provides application business object and service manager proxy, RPE, RPE proxy and service manager for serving as an intermediary between front end application program and service provider).

Brendle does not explicitly teach "a real time information director (RTID) that manages information flow between a data store and applications that supply and consume data, including transforming data under direction of polymorphic metadata".

However, Reed teaches "a real time information director (RTID) that manages information flow between a data store and applications that supply and consume data, including transforming data under direction of polymorphic metadata" (See Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 14, lines 56-60; col. 20, lines 55-59 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically carried out in real-time communication by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated, and data decryption and decompression).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Reed with Brendle reference

because both references are directed to providing sets of data elements and associated sets of operations where Reed teaches extending service-oriented business frameworks and processing business objects while Brendle teaches more on communication system for transferring data, metadata and methods from service provider to consumer, and the combined teaching of the references would have enabled Brendle's system to utilize Reed's metadata to define a control structure for communicating front end and back end components easily and quickly because communications as defined could have been automatically established and updated.

As per claim 33, Brendle teaches "A data services handler comprising" (See [0005] and [0049]-[0051] where data services handler includes application business object and service manager proxy, RPE, RPE proxy and service manager for serving as an intermediary between front end application program and service provider):

Brendle does not explicitly teach "a real time information director (RTID) that maps documents into a data model".

However, Reed teaches "a real time information director (RTID) that maps documents into a data model" (See Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 14, lines 56-60; col. 20, lines 55-59; col. 21, lines 32-36; col. 55, lines 8-10 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically carried out in real-time communication by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type

subclasses to which methods are associated, and fax message is transmitted via model or API interface).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Reed with Brendle reference because both references are directed to providing sets of data elements and associated sets of operations where Reed teaches extending service-oriented business frameworks and processing business objects while Brendle teaches more on communication system for transferring data, metadata and methods from service provider to consumer, and the combined teaching of the references would have enabled Brendle's system to utilize Reed's metadata to define a control structure for communicating front end and back end components easily and quickly because communications as defined could have been automatically established and updated.

The combined teaching of Reed and Brendle references further teaches the following:

"a disassembly logic that disassembles documents for loading into a database" (disassembling document and loading into database is an inherent feature of relational database, for example, Oracle® provides utility IMPort for loading documents into database);

"an assembly logic that assembles documents from records in the database" (See assembling document from database records is an inherent feature of relational database, for example, Oracle® provides extended SQL utility SQL-Plus for selecting records from database and generating report for the records); and

"a generator logic that generates an extended Markup Language (XML) Schema that describes format of the documents, the disassembly, assembly, and generator logic being operative using a single descriptive polymorphic metadata" (See Brendle: [0052], [0061] and [0088] where XML is utilized for exchanging information by importing and exporting tables, and Reed: Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 20, lines 55-59 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically carried out by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated).

As per claim 37, Brendle teaches "A content management system comprising" (See [0005] and [0049]-[0051] where content management system includes application business object and service manager proxy, RPE, RPE proxy and service manager for serving as an intermediary between front end application program and service provider):

Brendle does not explicitly teach "content metadata classes that describe a relationship between records and tables contributing to a document in an information handling system".

However, Reed teaches "content metadata classes that describe a relationship between records and tables contributing to a document in an information handling system" (See Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 20, lines 55-59; col. 24, lines 10-14 and col. 21, lines 32-36 where exchange of data,

metadata, methods and archiving and retrieval of changed objects can be automatically carried out by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated, and container classes provides elements for defining markup language output object, a document).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Reed with Brendle reference because both references are directed to providing sets of data elements and associated sets of operations where Reed teaches extending service-oriented business frameworks and processing business objects while Brendle teaches more on communication system for transferring data, metadata and methods from service provider to consumer, and the combined teaching of the references would have enabled Brendle's system to utilize Reed's metadata to define a control structure for communicating front end and back end components easily and quickly because communications as defined could have been automatically established and updated.

The combined teaching of Reed and Brendle references further teaches the following:

"a Structured Query Language (SQL) schema that corresponds to the content metadata and includes record definitions and field definitions" (See Reed: Fig. 3, element 141 and col. 20, lines 55-59 where method class as described in SQL schema is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated);

"a database arranged as native documents in an extended Markup Language (XML) representation" (See Brendle: [0052], [0061] and [0088] where XML is utilized for exchanging information by importing and exporting tables, and Reed: Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 20, lines 55-59; col. 24, lines 10-14 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically carried out by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated, and container classes provides elements for defining markup language output object, a document);
"SQL connections and prepared statements including document handlers, the database, document handlers, and metadata being arranged in parallel trees" (it is an inherent feature of database management system where SQL connections and prepared statement include document handlers, database and metadata in parallel, for example, when IMPort utility Oracle® database management system requires connection to database, opening and reading document records and insert records into database according to table metadata simultaneously).

As per claim 36, Brendle teaches "A method for generating native documents as an extended Markup Language (XML) representation of data in a Structured Query Language (SQL) database, the method comprising" (See Brendle: [0052], [0061] and

[0088] where XML is utilized for exchanging information by importing and exporting tables):

Brendle does not explicitly teach "creating metadata for a new native document further comprising".

However, Reed teaches "creating metadata for a new native document" (See Reed: Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 20, lines 55-59; col. 24, lines 10-14 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically carried out by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated, and container classes provides elements for defining markup language output object, a document).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Reed with Brendle reference because both references are directed to providing sets of data elements and associated sets of operations where Reed teaches extending service-oriented business frameworks and processing business objects while Brendle teaches more on communication system for transferring data, metadata and methods from service provider to consumer, and the combined teaching of the references would have enabled Brendle's system to utilize Reed's metadata to define a control structure for communicating front end and back end components easily and quickly because communications as defined could have been automatically established and updated.

The combined teaching of Reed and Brendle references further teaches the following:

"determining whether inbound documents are inserted, delserted or upserted" (See Reed: col. 29, lines 32-35 where communication objects all include functions of create, edit, delete and preview and whose combinations cover functions of determining inserted, delserted and upserted via previewing operation);

"determining what data from the SQL database is sufficient for usage in the new native document" (See Reed: Fig. 3, element 141 and col. 20, lines 55-59 where method class as described in SQL schema is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated);

"creating an XML Schema for the native document" (See Brendle: [0052], [0061] and [0088] where XML is utilized for exchanging information by importing and exporting tables, and Reed: Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 20, lines 55-59 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically carried out by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated).

As per claim 38, Brendle teaches "An article of manufacture comprising:

a controller-usable medium having a computable readable program code embodied

therein for generating native documents as an extended Markup Language (XML) representation of data in a Structured Query Language (SQL) database, the computable readable program code further comprising" (See Brendle: [0052], [0061] and [0088] where XML is utilized for exchanging information by importing and exporting tables):

Brendle does not explicitly teach "a code capable of causing the controller to create metadata for a new native document further comprising".

However, Reed teaches "a code capable of causing the controller to create metadata for a new native document" (See Reed: Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 20, lines 55-59; col. 24, lines 10-14 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically carried out by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated, and container classes provides elements for defining markup language output object, a document).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Reed with Brendle reference because both references are directed to providing sets of data elements and associated sets of operations where Reed teaches extending service-oriented business frameworks and processing business objects while Brendle teaches more on communication system for transferring data, metadata and methods from service provider to consumer, and the combined teaching of the references would have enabled

Brendle's system to utilize Reed's metadata to define a control structure for communicating front end and back end components easily and quickly because communications as defined could have been automatically established and updated.

The combined teaching of Reed and Brendle references further teaches the following:

"a code capable of causing the controller to determine whether inbound documents are delserted or upserted" (See Reed: col. 29, lines 32-35 where communication objects all include functions of create, edit, delete and preview and whose combinations cover functions of determining inserted, delserted and upserted via previewing operation);

"a code capable of causing the controller to determine what data from the SQL database is sufficient for usage in the new native document" (See Reed: Fig. 3, element 141 and col. 20, lines 55-59 where method class as described in SQL schema is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated);

"a code capable of causing the controller to create an XML Schema for the native document" (See Brendle: [0052], [0061] and [0088] where XML is utilized for exchanging information by importing and exporting tables, and Reed: Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 20, lines 55-59 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically carried out by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated).

As per claim 39, Brendle teaches "A data handling system comprising: means for generating native documents as an extended Markup Language (XML) representation of data in a Structured Query Language (SQL) database" (See Brendle: [0052], [0061] and [0088] where XML is utilized for exchanging information by importing and exporting tables):

Brendle does not explicitly teach "means for creating metadata for a new native document further comprising".

However, Reed teaches "means for creating metadata for a new native document" (See Reed: Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 20, lines 55-59; col. 24, lines 10-14 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically carried out by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated, and container classes provides elements for defining markup language output object, a document).

It would have been obvious to one having ordinary skill in the art at the time of the applicant's invention was made to combine the teaching of Reed with Brendle reference because both references are directed to providing sets of data elements and associated sets of operations where Reed teaches extending service-oriented business frameworks and processing business objects while Brendle teaches more on communication system for transferring data, metadata and methods from service

provider to consumer, and the combined teaching of the references would have enabled Brendle's system to utilize Reed's metadata to define a control structure for communicating front end and back end components easily and quickly because communications as defined could have been automatically established and updated.

The combined teaching of Reed and Brendle references further teaches the following:

"means for determining whether inbound documents are delserted or upserted" (See Reed: col. 29, lines 32-35 where communication objects all include functions of create, edit, delete and preview and whose combinations cover functions of determining inserted, delserted and upserted via previewing operation);

"means for determining what data from the SQL database is sufficient for usage in the new native document" (See Reed: Fig. 3, element 141 and col. 20, lines 55-59 where method class as described in SQL schema is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated);

"means for creating an XML Schema for the native document" (See Brendle: [0052], [0061] and [0088] where XML is utilized for exchanging information by importing and exporting tables, and Reed: Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 20, lines 55-59 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically carried out by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type

subclasses to which methods are associated).

As per claims 2 and 24, the combined teaching of Reed and Brendle references further teaches "a data mapper capable of collecting data from a plurality of diverse applications that can supply data in multiple diverse external document formats and mapping the supplied data to a predetermined unified data model according to the polymorphic metadata" (See Reed: col. 14, lines 46-59 where data, metadata and instructions in the API programs can control and automate the transfer and receipt of documents and method class is a form of metadata used to store method).

As per claim 3, the combined teaching of Reed and Brendle references further teaches "the real time information director comprises a logic that cleanses and enriches the data, supplies data security, and insures data integrity" (See Reed: Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 14, lines 56-60; col. 20, lines 55-59 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically carried out in real-time communication by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated).

As per claim 4, the combined teaching of Reed and Brendle references further teaches the following:

"a disassembler that disassembles inbound data documents into multiple records in preparation for insertion into metadata tables" (See Reed: col. 14, lines 46-59 where data, metadata and instructions in the API programs can control and automate the transfer and receipt of documents, including compression, decryption and decomposition of documents);
"an assembler that assembles outbound data documents from at least one record selected from at least one metadata-selected table" (See Reed: col. 14, lines 46-59 where data, metadata and instructions in the API programs can control and automate the transfer and receipt of documents, including compression, decryption and decomposition of documents).

As per claim 5, the combined teaching of Reed and Brendle references further teaches "a logic that defines a data model according to received descriptive metadata that identifies individual data elements and specifies logic behaviors applicable to the individual data elements" (See Reed: Fig. 3, element 141 and col. 20, lines 55-59 where method class is the metadata defining data elements and specifying logic behavior applying to the elements).

As per claim 30, the combined teaching of Reed and Brendle references further teaches "The enterprise system according to claim 28 further comprising "a server platform coupled to the real time information director that serves deployed models, customer data, and interaction data" (See Reed: Figs. 1, elements 33-35; Fig. 3,

element 141; col. 9, lines 45-49; col. 14, lines 56-60; col. 20, lines 55-59 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically carried out in real-time communication by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated).

As per claim 34, the combined teaching of Reed and Brendle references further teaches the following:

"content metadata that defines data to be included into a document" (See Brendle: [0052], [0061] and [0088] where XML is utilized for exchanging information by importing and exporting tables, and Reed: Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 20, lines 55-59 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically carried out by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated);

"format metadata that defines how the data appears in the document" (See Brendle: [0052], [0061] and [0088] where XML is utilized for exchanging information by importing and exporting tables, and Reed: Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 20, lines 55-59 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically

carried out by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated).

As per claim 6, the combined teaching of Reed and Brendle references further teaches "a logic that represents metadata as polymorphic objects, enabling individual metadata objects to have distinct numbers and types of attributes according to a predetermined object class, the behaviors and attributes being defined specific to object class" (See Reed: Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 14, lines 56-60; col. 20, lines 55-59 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically carried out in real-time communication by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated).

As per claim 31, the combined teaching of Reed and Brendle references further teaches "the RTID represents metadata as polymorphic objects, enabling individual metadata objects to have distinct numbers and types of attributes according to a predetermined object class, the behaviors and attributes being defined specific to object class" (See Reed: Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 14, lines 56-60; col. 20, lines 55-59 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically

carried out in real-time communication by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated).

As per claim 35, the combined teaching of Reed and Brendle references further teaches "a query logic that queries the database to determine information about tables in the database that the document uses, the information including field name, field datatype, and field allowance of nulls" (See Reed: Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 14, lines 56-60; col. 20, lines 55-59 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically carried out in real-time communication by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated).

As per claim 7, the combined teaching of Reed and Brendle references further teaches "a logic that represents metadata as stateless polymorphic objects that are permanent after initial construction, whereby methods of stateful metadata classes delegate operation to the stateless polymorphic metadata objects" (See Reed: Fig. 1, element 141 and col. 20, lines 50-59 where method class is a form of metadata for storing methods included in a communication object instance when transferred to a consumer).

As per claim 25, the combined teaching of Reed and Brendle references further teaches "a code capable of causing the controller to represent metadata as stateless polymorphic objects that are permanent after initial construction, whereby methods of stateful metadata classes delegate operation to the stateless polymorphic metadata objects" (See Reed: Fig. 1, element 141 and col. 20, lines 50-59 where method class is a form of metadata for storing methods included in a communication object instance when transferred to a consumer).

As per claim 32, the combined teaching of Reed and Brendle references further teaches "the RTID represents metadata as stateless polymorphic objects that are permanent after initial construction, whereby methods of stateful metadata classes delegate operation to the stateless polymorphic metadata objects" (See Reed: Fig. 1, element 141 and col. 20, lines 50-59 where method class is a form of metadata for storing methods included in a communication object instance when transferred to a consumer).

As per claim 8, the combined teaching of Reed and Brendle references further teaches "at least one metadata object constructed using constructors of multiple metadata classes, the constructors having parameters that specify metadata data elements and being nested to represent hierarchical relationships between the specified metadata data elements, the at least one metadata object including identifying

information for deriving the hierarchical relationships" (See Reed: Fig. 1, element 141 and col. 20, lines 50-59 where method class is a form of metadata for storing methods included in a communication object instance when transferred to a consumer).

As per claim 26, the combined teaching of Reed and Brendle references further teaches "a code capable of causing the controller to use constructor parameters to specify metadata data elements and being nested to represent hierarchical relationships between the specified metadata data elements, the at least one metadata object including identifying information for deriving the hierarchical relationships" (See Reed: Fig. 1, element 141 and col. 20, lines 50-59 where method class is a form of metadata for storing methods included in a communication object instance when transferred to a consumer).

As per claim 9, the combined teaching of Reed and Brendle references further teaches "a logic that uses received data document mapping metadata to disassemble elements of inbound data documents into rows and columns to be inserted into inbound metadata-selected tables in a database, select rows and columns from outbound metadata-selected tables of the database and assemble elements from the outbound metadata-selected tables into an outbound data document, and generate a schema for documents that can be input or output using the received data document mapping metadata" (See Reed: col. 14, lines 46-59 where data, metadata and instructions in the API programs can control and automate the transfer and receipt of documents,

including compression, decryption and decomposition of documents).

As per claim 27, the combined teaching of Reed and Brendle references further teaches "a code capable of causing the controller to use received data document mapping metadata to disassemble elements of inbound data documents into rows and columns to be inserted into inbound metadata-selected tables in a database, select rows and columns from outbound metadata-selected tables of the database and assemble elements from the outbound metadata-selected tables into an outbound data document, and generate a schema for documents that can be input or output using the received data document mapping metadata" (See Reed: col. 14, lines 46-59 where data, metadata and instructions in the API programs can control and automate the transfer and receipt of documents, including compression, decryption and decomposition of documents).

As per claim 10, the combined teaching of Reed and Brendle references further teaches "the real time information director receives and publishes data documents having a content and a format defined by document metadata" (See Reed: col. 14, lines 46-59 where data, metadata and instructions in the API programs can control and automate the transfer and receipt of documents, including compression, decryption and decomposition of documents, and the control is executed in real-time).

As per claim 11, the combined teaching of Reed and Brendle references further teaches the following:

"a logic that reads a schema of tables identified in metadata" (reading tables from dictionary or catalog is inherent to a database system, for example, a SQL statement allows reading every table of an identified user);

"a logic that automatically generates and caches prepare SQL statements from a content metadata portion of document metadata according to the schema" (See Reed: col. 72, lines 20-27 where query statement returns result for appending communication object).

As per claim 12, the combined teaching of Reed and Brendle references further teaches the following:

"a servlet that invokes the real time information director synchronously when an inbound data document is delivered via HyperText Transport Protocol (HTTP)" (See Reed: col. 8, lines 40-44 and col. 14, lines 46-59 where information is created and maintained according to a recognized protocol, such as HTTP and data, metadata and instructions are transferred in real-time control);

"a logic that invokes the real time information director asynchronously when an inbound data document is delivered via Java Message Service (JMS)" (See Brendle: [0072] where communication channel is implemented in Java Connector for specifying parameters);

"a servlet that invokes the real time information director synchronously when an

outbound data document is requested via HyperText Transport Protocol (HTTP)" (See Reed: col. 16, lines 28-32 and col. 15, lines 34-35 where applets are implemented in Java language for providing additional interface functions and consumer program generates request in HTTP protocol);
"a logic that invokes the real time information director asynchronously when an outbound data document is requested via Java Message Service (JMS)" (See Brendle: [0072] where communication channel is implemented in Java Connector for specifying parameters).

As per claim 13, the combined teaching of Reed and Brendle references further teaches "a subscription handler that responds to a triggering event by publishing a document according to content and format specified by Document Metadata, and sending the document to a designated queue or topic" (See Reed: col. 30, lines 59-67 where archiving rule is triggered each time an edited form is submitted and the rule determined if the document needs to be deleted and updated with a new version).

As per claim 14, the combined teaching of Reed and Brendle references further teaches the following:
"a security manager that identifies an agent on behalf of whom information is queried or updated in a document request, dynamically assigns an internal unique identifier to consumers and agents upon registration" (See Reed: col. 68, lines 20-28 where data security control is implemented for data exchange in communication objects system);

"the real time information director grants access to individual documents based on security policies according to a relationship between an agent, a consumer affected by the agent, and permissions granted by the consumer" (See Reed: col. 68, lines 20-28 where data security control is implemented for data exchange in communication objects system).

As per claim 15, the combined teaching of Reed and Brendle references further teaches "the security manager enforces securities policies that are specifiable for each document" (See Reed: col. 68, lines 20-28 where data security control is implemented for data exchange in communication objects system).

As per claim 16, the combined teaching of Reed and Brendle references further teaches "the security manager enforces securities policies that are customized to identify specific agent and consumer relationships expressed in a business model" (See Reed: col. 68, lines 20-28 where data security control is implemented for data exchange in communication objects system).

As per claim 17, the combined teaching of Reed and Brendle references further teaches "an audit log handler that responds to changes in the data store by compiling and storing information based on metadata that determines content and detail of the compiled information" (See Reed: col. 14, lines 46-59 where data, metadata and instructions transferred between provider and consumer programs are formatted,

compressed, encrypted; and decrypted, decompressed and acknowledged on receipt, respectively).

As per claim 18, the combined teaching of Reed and Brendle references further teaches "a data cleanser that receives data elements identified by Document Metadata, normalizes the data elements, queues source and destination information for detection of duplicates, and cross-references duplicates in the data store" (See Reed: col. 14, lines 46-59 where data, metadata and instructions transferred between provider and consumer programs are formatted, compressed, encrypted; and decrypted, decompressed and acknowledged on receipt, respectively).

As per claim 19, the combined teaching of Reed and Brendle references further teaches "an aggregation engine that computes aggregates in memory from data selected from the data store" (See Reed: Fig. 3, col. 14, lines 46-59 where data, metadata and instructions transferred between provider and consumer programs are formatted, compressed, encrypted; decrypted, decompressed and acknowledged on receipt, and communication objects class types includes aggregation type suggests data aggregation).

As per claim 20, the combined teaching of Reed and Brendle references further teaches "a real-time scoring engine" (See Reed: col. 14, lines 57-59 where control

techniques is applied to real-time communications suggests the teaching of real-time scoring).

As per claim 21, the combined teaching of Reed and Brendle references further teaches "a business rules engine that defines a business object model capable of enabling access to data described by Document Metadata, performing computations on data relevant to business rules specified by the business object model, concluding a decision based on computation results, and executing the decision" (See Brendle: [0005] and [0029]-[0034] where business objects and processing are built on a meta data model, Reed: Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 14, lines 56-60; col. 20, lines 55-59 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically carried out in real-time communication by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated).

As per claim 23, the combined teaching of Reed and Brendle references further teaches "a code capable of causing the controller to transform data under direction of polymorphic metadata that defines a security model and data integrity rules for application to the data" (See Reed: Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 14, lines 56-60; col. 20, lines 55-59 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects

can be automatically carried out in real-time communication by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated).

As per claim 29, the combined teaching of Reed and Brendle references further teaches "a business rules engine coupled to the real time information director that personalizes transactions by applying business rules and prediction models against cached entity data" (See Brendle: [0005] and [0029]-[0034] where business objects and processing are built on a meta data model, Reed: Figs. 1, elements 33-35; Fig. 3, element 141; col. 9, lines 45-49; col. 14, lines 56-60; col. 20, lines 55-59 and col. 21, lines 32-36 where exchange of data, metadata, methods and archiving and retrieval of changed objects can be automatically carried out in real-time communication by provider/consumer programs and method class is a form of metadata of specialized types, and method types can be determined by a type attribute or by method type subclasses to which methods are associated).

Conclusion

8. The prior art made of record

A. U.S. Patent No. 6,088,717

B. U.S. Patent Application 2005/0021354

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

C. U.S. Patent Application 2005/0091276

D. U.S. Patent No. 7,062,500

E. U.S. Patent Application 2005/0021998

F. U.S. Patent Application 2005/0044165

G. U.S. Patent Application 2004/0045014

Contact Information

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kuen S Lu whose telephone number is (571) 272-4114. The examiner can normally be reached on Monday-Friday (8:00 am-5:00 pm). If attempts to reach the examiner by telephone pre unsuccessful, the examiner's Supervisor, John Cottingham can be reached on (571) 272-7079. 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 Page 13 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 886-217-9197 (toll-free).

Kuen S. Lu

Patent Examiner, Art Unit 2167

November 28, 2006

PATENT APPLICATION

Sheet 1 of 1

FORM PTO-1449 LIST OF PATENTS AND PUBLICATIONS FOR APPLICANT'S INFORMATION DISCLOSURE STATEMENT (Use several sheets if necessary)	ATTY. DOCKET NO. 200400406-1	APPLICATION NO. 10/868,434	CONFIRMATION NO.
	APPLICANT Steven R. Carr et al.		
	FILING DATE June 14, 2004	GROUP 2167	

REFERENCE DESIGNATION **U.S. PATENT DOCUMENTS**

EXAMINER INITIAL		DOCUMENT NUMBER	PUBLICATION DATE	NAME	Pages, Columns, Lines Where Relevant Passages or Figures Appear
KL	1A	6,157,707	12-05-2000	Baulier et al.	
KL	1B	6,163,604	Dec. 19, 2000	Baulier et al.	
KL	1C	6,233,575 B1	May 15, 2001	Agrawal et al.	
KL	1D	6,240,411 B1	May 29, 2001	Thearling	
KL	1E	6,317,700 B1	Nov. 13, 2001	Bagne	
KL	1F	6,480,844 B1	Nov. 12, 2002	Cortes et al.	
KL	1G	6,484,163 B1	Nov. 19, 2002	Lawrence et al.	
KL	1H	6,519,576 B1	Feb. 11, 2003	Freeman	
KL	1I	6,557,008 B1	Apr. 29, 2003	Temple, III et al.	
KL	1J	6,571,279 B1	May 27, 2003	Herz et al.	
	1K				

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	PUBLICATION DATE	NAME OF PATENTEE OR APPLICANT	Pages/Columns/Lines Where Relevant Passages/Figures Appear	Check if Translation attached
	1L					
	1M					
	1N					
	1O					
	1P					

OTHER REFERENCES (including Author, Title, Date, Pertinent Pages, etc.)

	1Q	
	1R	
	1S	

EXAMINER	DATE CONSIDERED
/Kuen Lu/ (11/25/2006)	/Kuen Lu/ (11/25/2006)

Rev 05/04 (PTO1449)

Notice of References Cited	Application/Control No. 10/868,434	Applicant(s)/Patent Under Reexamination CARR ET AL.	
	Examiner Kuen S. Lu	Art Unit 2167	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,088,717	07-2000	Reed et al.	709/201
*	B	US-2005/0021354	01-2005	Brendle et al.	705/001
*	C	US-2005/0091276	04-2005	Brunswig et al.	707/104.1
*	D	US-7,062,500	06-2006	Hall et al.	707/102
*	E	US-2005/0021998	01-2005	Fiedler et al.	713/200
*	F	US-2005/0044165	02-2005	O'Farrell et al.	709/213
*	G	US-2004/0045014	03-2004	Radhakrishnan, Rakesh	719/328
	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	
	10/868,434	CARR ET AL.	
	Examiner	Art Unit	
	Kuen S. Lu	2167	

SEARCHED			
Class	Subclass	Date	Examiner
707	104.1	11/28/2006	KLU
705	1	11/28/2006	KLU
709	201	11/28/2006	KLU
707	100	11/28/2006	KLU

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)		
	DATE	EXMR
consult with Hung Pham, Examiner of Art Unit 2168, concerning potential 35 USC 101 issues	11/25/2006	KLU
EAST(11/25-28/2006)	11/28/2006	KLU

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. 4452

SERIAL NUMBER	FILING OR 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
10/868,434	06/14/2004 RULE	707	2167	200400406-1

APPLICANTS
Steven R. Carr, San Jose, CA;
Venkatakrishnan Muthuswamy, Cupertino, CA;
Tudor Har, Santa Clara, CA;

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

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

IF REQUIRED, FOREIGN FILING LICENSE GRANTED
** 08/12/2004

Foreign Priority claimed	STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
☐ yes ☑ no	CA	24	39	8
35 USC 119 (a-d) conditions met ☐ yes ☑ no ☐ Met after Allowance				
Verified and Acknowledged Examiner's Signature Initials				

ADDRESS
22879

TITLE
Data services handler

FILING FEE RECEIVED	
1542	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

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 162 of 302

Index of Claims

Application/Control No.	Applicant(s)/Patent under Reexamination
10/868,434	CARR ET AL.
Examiner	**Art Unit**
Kuen S. Lu	2167

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	+	Restricted	I	Interference	O	Objected

Claim		Date
Final	Original	11/25/06
	1	√
	2	√
	3	√
	4	√
	5	√
	6	√
	7	√
	8	√
	9	√
	10	√
	11	√
	12	√
	13	√
	14	√
	15	√
	16	√
	17	√
	18	√
	19	√
	20	√
	21	√
	22	√
	23	√
	24	√
	25	√
	26	√
	27	√
	28	√
	29	√
	30	√
	31	√
	32	√
	33	√
	34	√
	35	√
	36	√
	37	√
	38	√
	39	√
	40	
	41	
	42	
	43	
	44	
	45	
	46	
	47	
	48	
	49	
	50	

Claim		Date
Final	Original	
	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	

Claim		Date
Final	Original	
	101	
	102	
	103	
	104	
	105	
	106	
	107	
	108	
	109	
	110	
	111	
	112	
	113	
	114	
	115	
	116	
	117	
	118	
	119	
	120	
	121	
	122	
	123	
	124	
	125	
	126	
	127	
	128	
	129	
	130	
	131	
	132	
	133	
	134	
	135	
	136	
	137	
	138	
	139	
	140	
	141	
	142	
	143	
	144	
	145	
	146	
	147	
	148	
	149	
	150	

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 163 of 302

EAST Search History

10/868,434

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L2	4	707/104.1.ccls. and (service with manager with application with (database or (data adj source))) and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/28 15:44
L3	4	705/1.ccls. and (service with manager with application with (database or (data adj source))) and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/28 15:43
L4	2	707/100.ccls. and (service with manager with application with (database or (data adj source))) and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/28 15:44
L5	1	709/201.ccls. and (service with manager with application with (database or (data adj source))) and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/28 15:44
S1	2	("20050278270").PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/26 14:37
S2	0	("polymorphicadjmeta$1data").PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/25 16:50
S3	2	polymorphic adj meta$1data	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/25 16:51
S4	2	polymorphic with meta$1data	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/25 16:51
S5	183	polymorphic and meta$1data	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/25 16:51

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 164 of 302

EAST Search History

S6	19	polymorphic and meta$1data and (security with model)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/25 16:52
S7	16	polymorphic and meta$1data and (security with model) and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/25 16:52
S8	16	polymorphic and meta$1data and (security with model) and @ad < "20040614" and interface	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/25 16:53
S9	0	polymorphic and meta$1data and (security with model) and @ad < "20040614" and interface and (data with intergrity)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/25 16:53
S10	6	polymorphic and meta$1data and (security with model) and @ad < "20040614" and interface and (data with integrity)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 10:42
S11	63	mid$4ware and meta$1data and (security with model) and @ad < "20040614" and interface and (data with integrity)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 10:42
S12	1	(mid$4ware with service with manager) and meta$1data and (security with model) and @ad < "20040614" and interface and (data with integrity)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 10:47
S13	0	(service with manager with meta$1data) and (security with model) and @ad < "20040614" and interface and (data with integrity)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 10:47
S14	3	(service with manager with application with (database or (data adj source))) and meta$1data and (security with model) and @ad < "20040614" and interface and (data with integrity)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 12:46

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 165 of 302

EAST Search History

S15	51	(service with manager with application with (database or (data adj source))) and meta$1data and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/28 15:43
S16	0	(service with manager with application with (database or (data adj source))) and (meta$1data with types with attributes) and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 12:47
S17	37	(service with manager with application with (database or (data adj source))) and (meta$1data and types and attributes and class and object) and @ad < "20040614"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 12:48
S18	0	(service with manager with application with (database or (data adj source))) and (meta$1data and types and attributes and class and object) and @ad < "20040614" and polymorhic	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 12:49
S19	11	(service with manager with application with (database or (data adj source))) and (meta$1data and types and attributes and class and object) and @ad < "20040614" and polymorphic	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 14:27
S20	0	(service with manager with application with (database or (data adj source))) and (meta$1data and types and attributes and class) and @ad < "20040614" and (polymorphic with object)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 12:49
S21	0	(service with manager with application with (database or (data adj source))) and (meta$1data and types and attributes and class) and @ad < "20040614" and (polymorphic same object)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 12:49
S22	1	(meta$1data or polymorphic) and "20050021354"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 14:27

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 166 of 302

EAST Search History

S23	0	("6088717.pn.").PN.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/11/26 14:36
S24	2	"6088717".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 14:36
S25	2	"20050027871"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/26 14:38
S26	4	"20050021354" or "6088717".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/11/27 10:09

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 167 of 302

RECEIVED
CENTRAL FAX CENTER

JUN 14 2006

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200400406-1

Inventor(s): Steven Robert Carr et al

Application No.: 10/868434

Filing Date: Jun 14, 2004

Confirmation No.: 4482

Examiner: Unknown

Group Art Unit: 2182

Title: Data Services Handler

Commissioner For Patents
PO Box 1450
Alexandria, VA 22313-1450

STATUS INQUIRY

More than 24 months have passed since:

☒ NEW APPLICATIONS
The filing of this application on 14 June 2004. No communication has been received from the Patent and Trademark Office indicating action on this application.

☐ AMENDED APPLICATIONS
The filing of a response on _____. No communication has been received from the Patent and Trademark Office.

☐ APPEALED APPLICATION
The Appeal Brief was filed on _____.

☐ An Examiner's Answer was mailed on _____.

☐ A Reply Brief to the Examiner's Answer was submitted on _____.

☐ ALLOWED APPLICATIONS
The mailing of FORM POL-85 and/or Examiner's Amendment on _____.

Kindly advise the undersigned of the present status of this application.

☐ I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450

Date of Deposit:

OR

☒ I hereby certify that this paper is being transmitted to the Patent and Trademark Office facsimile number (571) 273-8300.
Date of facsimile: 14 June 2006

Typed Name: Debi Bonacquisti

Signature: Debi Bonacquisti

Respectfully submitted,

By _____
Attorney/Agent for Applicant(s)

Gerald E. Laws

Reg No.: 38,258

Date: 14 June 2006

Telephone: 281 518 7159

Rev 10/05 (Status)

23026 U.S.PTO
061404

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200400406-1

IN THE U.S. PATENT AND TRADEMARK OFFICE
Patent Application Transmittal Letter

22141 U.S.PTO
10/868434
061404

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

Transmitted herewith for filing under 37 CFR 1.53(b) is a(n): (X) Utility ( ) Design
(X) original patent application,
( ) continuation-in-part application

INVENTOR(S): **Steven R. Carr et al.**

TITLE: **DATA SERVICES HANDLER**

Enclosed are:

(X) The Declaration and Power of Attorney. (X) signed ( ) unsigned or partially signed

(X) 24 sheets of drawings (one set) ( ) Associate Power of Attorney

( ) Form PTO-1449 (X) Information Disclosure Statement and Form PTO-1449

( ) Priority document(s) (X) (Other) - Application Data Sheet (fee $_____)
- Recordation Form Cover Sheet
- Assignment

CLAIMS AS FILED BY OTHER THAN A SMALL ENTITY				
(1) FOR	(2) NUMBER FILED	(3) NUMBER EXTRA	(4) RATE	(5) TOTALS
TOTAL CLAIMS	39 — 20	19	X $18	$ 342
INDEPENDENT CLAIMS	8 — 3	5	X $86	$ 430
ANY MULTIPLE DEPENDENT CLAIMS	0		$290	$ 0
BASIC FEE: Design ($340.00); Utility ($770.00)				$ 770
TOTAL FILING FEE				$ 1,542
OTHER FEES				$
TOTAL CHARGES TO DEPOSIT ACCOUNT				$ 1,542

Charge $ 1,542 to Deposit Account 08-2025. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

"Express Mail" label no. EU 959884178 US

Date of Deposit June 14, 2004

I hereby certify that this is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to: Commissioner for Patents, Alexandria, VA 22313-1450.

By _____

Typed Name: Joy C. Ngo

Respectfully submitted,

Steven R. Carr et al.

By _____

Ken J. Koestner

Attorney/Agent for Applicant(s)
Reg. No. 33,004

Date: June 14, 2004

Telephone No.: (949) 251-0250

Rev 05/04 (TransNew)

- Attach as First Page to Transmitted Papers -

HPDNO.: 200400406-1

DATA SERVICES HANDLER

Steven R. Carr
3115 Terra Cotta Ct.
San Jose, CA 95135
Citizenship: USA

Venkatakrishnan Muthuswamy
10870 N. Stelling Rd. #25A
Cupertino, CA 95014
Citizenship: INDIA

Tudor Har
2541 South Park Lane
Santa Clara, CA 95051
Citizenship: USA

Docket No. 200400406-1
KB No.: 1015.P098 US

EXPRESS MAIL LABEL NO:
EU959884178US

DATA SERVICES HANDLER

Steven R. Carr
Venkatakrishnan Muthuswamy
Tudor Har

BACKGROUND OF THE INVENTION

[0001] Enterprises today are driven to improve responsiveness to customers and events, an objective commonly sought through usage of a computing infrastructure that attains access to integrated, real-time information. One aspect of real-time enterprises is a capability to materialize data from disjointed databases into an integrated data model. Many problems, such as synchronization of master data, cannot be solved effectively by a federated database approach alone. Data elements are deduplicated and merged into a common data model to create a semantic bridge between systems. Existing tools and methods operate in batch mode to support business intelligence applications, but little is available to support real-time information integration.

[0002] Latency, an inability to react immediately to business stimuli, can contribute to basically any deficiency in business performance, for example poor customer service, missed selling opportunities, failure to address consumer fraud, insufficiency in monitoring enterprise finances, and the like. A zero latency enterprise (ZLE) performance has been sought to enable enterprises to eliminate latency from operations so that business events that occur anywhere in an organization can immediately trigger appropriate actions across other parts of the enterprise and beyond. Elimination of operational inconsistency is sought to enable users to gain real-time consolidated performance and enable the enterprise to become more responsive and competitive.

[0003] The challenge is attaining a zero latency performance so that a business can integrate, synchronize, and route data across the enterprise, all in real time.

[0004] One technique that has been attempted to attain a low latency performance relies on standard applications to supply data services using a standard data model. Another technique attempts to customize a data model without recoding applications through usage of table-driven code. Table-driven code is deficient in capability to handle new data cases well if the new cases have attributes unlike previous cases. Another attempted solution is usage of user-exits to evade limitations of table-driven approaches. User-exits are difficult to integrate into the application, as data and functions available to user-exits are typically very limited.

[0005] While many industry participants have sought real-time performance, few have attained more than simple asynchronous message passing middleware. Similarly, an adaptive computing capability has been discussed but not progressed beyond simple system level concepts such as computing on demand.

SUMMARY

[0006] In accordance with an embodiment of a data services system, a data services handler comprises an interface between a data store and applications that supply and consume data, and a real time information director (RTID) that transforms data under direction of polymorphic metadata that defines a security model and data integrity rules for application to the data.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Embodiments of the invention relating to both structure and method of operation, may best be understood by referring to the following description and accompanying drawings whereby:

FIGUREs 1A and **1B** are schematic block diagrams that depict an embodiment of a data services handler;

FIGUREs 2A and **2B** are schematic graphical diagrams illustrating an embodiment of a Real Time Supply Chain (RTSC) for usage in conjunction with the data services handler;

FIGUREs 3A and **3B** are schematic block diagrams illustrating embodiments of a data services handler that comprises a real time information director (RTID) capable of mapping documents into a data model;

FIGURE 4 is a schematic block diagram showing an embodiment of a data handling system in a configuration suitable for handling subscriptions to documents;

FIGUREs 5A and **5B** illustrate embodiments of data services handlers that perform an identity management function and multiple-level security functionality, respectively;

FIGURE 6 is a schematic block diagram illustrating an embodiment of a data services handler that is configured with a data auditing capability;

FIGUREs 7A and **7B** are schematic block diagrams illustrating a data services handler capable of implementing a data cleansing functionality;

FIGURE 8 is a schematic block diagram showing an embodiment of a data services handler that implements business rules handling;

FIGURE 9 is a schematic block diagram illustrating an embodiment of a data services handler that operates according to a request-response model;

FIGUREs 10A through **10K** are schematic block diagrams illustrating embodiments of various real time information director interfaces;

FIGURE 11 is a schematic block diagram illustrating an embodiment of an information handling system that can be used for Zero Latency Enterprise (ZLE) applications;

FIGUREs 12A through **12E** are Unified Modeling Language (UML) diagrams that describe operations and functionality of the real time information director;

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 173 of 302

FIGURE 13 is a Unified Modeling Language (UML) diagram that illustrates various enrichment classes; and

FIGURE 14 is a UML diagram illustrating an embodiment of IDoc format metadata that depicts a technique for implementing format metadata for Systems, Applications, and Products (SAP) IDocs.

DETAILED DESCRIPTION

[0008] An example of a real-time infrastructure is Supply Chain™ from Hewlett-Packard of Palo Alto, California, which operates in a distributed fashion, with geographies and business units having different business processes. To respond to customers and suppliers more effectively, a real-time computing infrastructure is implemented which maps data from many of the hundreds of Systems, Applications, and Products (SAP) instances into a common data model, embodied in a real-time data store. SAP of Walldorf, Germany supplies enterprise systems. The result is a unified view of the Supply Chain, up to the current minute, that attains an ability to manage supply globally, enabling more efficient pricing, supplier contracts, and inventory turns.

[0009] The illustrative embodiments can be implemented in a real-time computing architecture with a capability to supply a solution that makes an enterprise more adaptive as a business by supplying a capability to integrate disparate enterprise applications in real time, addressing business-level flexibility and timeliness. An enterprise system is a computing environment or model that includes multiple interlinked servers and networks in which multiple users access applications and data stored on a server.

[0010] A real-time infrastructure has one or more of several desirable characteristics including an extreme adaptability of the data model, allowing rapid customization. In illustrative embodiments, relationships between source data and a central data model are described as metadata. Other characteristics include support for cleansing, enrichment, integration, and aggregation of data for use by subscribing applications, a capability for mapping and translating data and messages from numerous structures and database technologies into a common data model. Data cleansing, which may otherwise be termed data scrubbing, is the process of amending or removing data in a database that is

incomplete, incorrect, improperly formatted, or duplicated. In a data-intensive application such as telecommunications, transportation, banking, insurance, or retailing, data cleansing is used to systematically examine data for flaws using rules, algorithms, data relationships, such as look-up tables, and the like. Other characteristics include a simple and open subscription service, security and integrity of data in the real-time data store, and a capability for synchronous and asynchronous delivery of inbound and outbound data.

[0011] An integrated set of services called a real time information director (RTID) has been developed to address performance and functionality of a real-time enterprise. The real time information director is responsible for most or all data input and output to the real-time data store. The real time information director receives and sends data in the form of eXtended Markup Language (XML) documents. In an illustrative embodiment, the real time information director is implemented in the Java language and runs through WebLogic Server™ from BEA of San Jose, California. The real time information director is accessed synchronously via HyperText Transport Protocol (HTTP) and asynchronously via Java Message Service (JMS). Hewlett-Packard's OpenView™ manages the real time information director through standard Java 2 Enterprise Edition (J2EE) and NonStop WebLogic Server (WLS) mechanisms. The real time information director interoperates with various Enterprise Application Integration (EAI) platforms and middleware, such as SAP Netweaver XI™ from SAP of Walldorf, Germany, various programs from Web Methods™ of Fairfax, Virginia, Rendezvous™ from TIBCO Software, Inc. of Palo Alto, California, BEA eLink™ and HP Service Bus™. The EAI platforms supply adapters to many other applications that feed or consume data from the real-time data store.

[0012] The real time information director prepares and caches Java Database Connectivity (JDBC) statements for each inbound or outbound document class. A "document disassembler" contains prepared statements for inserting or updating various records related to a particular class of inbound documents into the data store. A "document assembler" contains prepared statements for selecting various records related to a particular class of outbound documents from the data store. The real time information director can instantiate and cache for reuse as many disassemblers and assemblers of each document class as is appropriate to handle peak concurrent demand.

[0013] Each class of documents is defined through descriptive metadata. Assemblers and disassemblers are created on-the-fly, based entirely on the descriptive metadata and on the schema of the tables read on demand from the database. Because the metadata is descriptive rather than procedural, the same metadata is used to build both assemblers and disassemblers of the same document class.

[0014] Metadata is expressed in Java, mainly with nested Java constructors. New Java metadata subclasses are created as appropriate to handle novel criteria. The metadata classes extensively use polymorphism. Format metadata classes map documents from externally-defined representations, such as SAP IDocs, to a common data model. Content metadata classes describe the relationships between the various records and tables that contribute to a particular document. Other metadata classes invoke various data services and cleanse or enrich the document, either before insertion into the real-time data store, or when assembling an outbound document.

[0015] A zero latency enterprise (ZLE) data store is developed that is capable of consolidating data that resides in multiple applications.

[0016] A real time information director is developed that supplies multiple services for clients of the ZLE data store including security, auditing, data cleansing, deduplication, aggregation, real-time scoring, business rules, accumulators, user-written functions, and the like.

[0017] The real time information director uses metadata, defined as data that describes various aspects of other data, for multiple purposes. In various embodiments, the real time information director uses the same descriptive metadata for purposes including controlling disassembly of documents for loading into a database, controlling assembly of documents from records in the database, generating an eXtended Markup Language (XML) schema that describes the document format, and other purposes.

[0018] Descriptive metadata is typically very concise in comparison to procedural code. For example, real time information director (RTID) metadata for individual SAP IDocs is typically about one-tenth the size of the eXtensible Stylesheet Language

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 176 of 302

Transformations (XSLT) style sheets that have been created to map the IDoc to the Real-Time Supply Chain (RTSC) data model.

[0019] Polymorphic metadata can be used to enable extension of the real time information director to facilitate and address new functionality and behaviors. Polymorphic metadata is metadata that takes many forms, using polymorphism to flexibly represent different patterns economically and facilitate evolution of metadata as new circumstances arise. Generally, an individual metadata class can be used to represent a different common pattern, defined with attributes that are specific to the pattern. For example, circles and rectangles can be assigned to different classes of Shape with Circle having a defining attribute called diameter, and Rectangle having defining attributes called length and width. The polymorphic representation is economical because specified attributes or parameters of metadata can be limited to those that are applicable to a specific pattern under consideration. New metadata classes can be rapidly generated to solve new problems outside of the patterns previously encountered.

[0020] In some embodiments, polymorphic metadata can be expressed in Java, enabling usage of descriptive language for new classes of metadata to be implemented at the same time as the behavior that carries out that metadata. The descriptive power of the metadata can be embodied in Java constructors.

[0021] In a Java implementation, individual classes of real time information director (RTID) metadata can be implemented as distinct Java classes. Parameters passed to the constructors for each metadata class serve to describe the metadata. Other methods of metadata classes are called to carry out the functions of the metadata. A new class of metadata is most frequently created by creating a new subclass of an existing metadata class. The subclass can be used in any context in which the superclass could be used. Additional parameters can be defined for the sub-class, which can then be used within the implementation of any methods that override the inherited behaviors of the superclass. The metadata classes are stateless and are called by the core functionality of the real time information director (RTID) whenever the real time information director is operating on relevant data. Stateful data objects are passed to the relative metadata methods as parameters.

[0022] Referring to **FIGURE 1A**, a schematic block diagram depicts an embodiment of a data services handler **100** that comprises an interface **102** between a data store **104** and applications, shown as applications **106**, which supply and consume data, and a real time information director **108**. The real time information director **108** transforms data under direction of polymorphic metadata that defines a security model and data integrity rules for application to the data.

[0023] In a particular embodiment, the data services handler **100** can be implemented according to a Zero Latency enterprise (ZLE) architecture or framework, a multiple-level architecture centered on a virtual hub or ZLE core **110**, that caches and routes information. Contained within the ZLE core **110** are persistent ZLE data store **104** and various components or devices that maintain the data store **104** in synchrony with various systems integrated by the core **110**. ZLE supports many enterprise applications, for example including Systems, Applications, and Products (SAP) in data processing, Siebel Systems, and other enterprise systems, that attach to the ZLE core **110** via a variety of application and technology adapters, shown as adapters **112**. The adapters **112** convert proprietary messages to standards supported by the ZLE core **110**. Siebel Systems, Inc., of San Mateo, California, supplies Customer Relationship Management (CRM) and business intelligence systems.

[0024] ZLE data store **104** caches information passing in and out of the core **110**, maintains and enriches the information, and operates as an intelligent message store and queue to manage information delivery to the appropriate applications. Data in the ZLE data store **104** can be mined and used for business intelligence purposes, either through bulk extracts to data mining tools and data markets, or by direct queries against the data store **104**. Data mining and other knowledge discovery applications that retrieve data from the data store **104** can also contribute models back to the ZLE core **110**, where the models can be leveraged to direct business processes.

[0025] Specialized ZLE applications, such as personalization and fraud management, reside above the ZLE core **110** to directly access data in real-time and leverage real-time integration and other services.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 178 of 302

[0026] Real time information director **108** supplies data services for various clients of a ZLE data store **104**, or other Operational Data Store, including both data providers and data consumers. Real time information director **108** ensures data integrity and data security that can be particularly useful because no single application has exclusive access to data in the ZLE data store **104**. Real time information director **108** supports a customizable data model that is not constrained to a particular program code, through usage of polymorphic metadata.

[0027] Referring to **FIGURE 1B**, a schematic block diagram illustrates an embodiment of a data services handler **100** that uses the ZLE data store **104** as a physical and logical center of the ZLE core **110**. The ZLE data store **104** can be based on an operational data store (ODS) model type that functions as a database front end for new and existing applications attached to the ZLE core **110**. A conventional data warehouse is typically a back-end database constructed for after-the-fact analysis functionality. In contrast, the ZLE data store **104** resides logically in front of other applications **106** in the ZLE environment that uses the data store **104** as basis for real-time integration. The ZLE data store **104** accordingly operates as a hot enterprise cache, a state engine, a real-time data warehouse, and a relational message store.

[0028] By caching data in the ZLE data store **104**, the ZLE framework enables applications and systems to be loosely coupled, improving over conventional closely-integrated systems which are limited in terms of scalability and availability. Specifically, conventional systems typically cannot function without all integrated systems being currently connected, or cannot scale independently. For example, a conventional e-store may be constrained to access a credit system to check a customer rating before completing a transaction. The ZLE framework caches data in a central repository to loosen application bonds without disrupting integration.

[0029] As a hot enterprise cache, ZLE data store **104** can update operational data continuously in real-time. Continuous database availability is enabled in some embodiments through usage of a NonStop SQL database in the ZLE core **110**. The NonStop SQL database currently supports most international stock exchanges, the majority of Point Of Sale (POS) and Asynchronous Transfer Mode (ATM) networks, and other critical large-scale applications that cannot risk downtime. The NonStop SQL

database has characteristics of massive linear scalability, parallelism, and continuous availability.

[0030] The ZLE data store **104** can further function as a state engine that represents the business state of an enterprise, a single location at which the state of a company is defined at a particular moment. A typical function of a database is online transaction processing (OLTP), operating as a state engine to track long-running applications. The single, consolidated database formed by the ZLE data store **104** enables development of new ZLE applications that reside above the data store **104** and leverage the current state of the business view. The applications can include new types of applications to manage customer relationships, prevent or reduce fraud, and the like.

[0031] Business state information is made available to many different systems enabling the database to function as a focal point of activity for simultaneous updating as transactional systems send updates to the data store **104**, accessing by customer service applications, and querying and extraction to other operational and analytic systems.

[0032] ZLE data store **104** can also function as a real-time data warehouse for feeding downstream users subsets of real-time data. ZLE data store **104** can function as a real-time data warehouse at least partly on the basis that the NonStop SQL database resolves differing operating constraints of OLTP and business intelligence.

[0033] ZLE data store **104** can also function as a relational message store that is useful in a publish/subscribe environment. Messages moving through the ZLE core **110** can be maintained in a relational database that forms a basis for the ZLE data store **104**. Individual messages, for example representing a POS transaction, a call detail report, a credit or debit transaction, or the like, can be stored and managed in a single, centralized location, enabling routing to multiple subscriber systems.

[0034] A publish/subscribe feature can be implemented in the NonStop SQL database to enable subscriptions to be created in SQL as select statements with output data streamed to the subscribing application. Subscriptions can be installed into tables, enabling waiting for events to occur.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 180 of 302

[0035] Through the NonStop SQL database, publish/subscribe functionality enables messaging to leverage the database platform parallelism, partitioning, and manageability to ensure efficient flow of subscriber messages.

[0036] The various functions, processes, methods, and operations performed or executed by the system can be implemented as programs that are executable on various types of servers, processors, controllers, central processing units, microprocessors, digital signal processors, state machines, programmable logic arrays, and the like. The programs can be stored on any computer-readable medium for use by or in connection with any computer-related system or method. A computer-readable medium is an electronic, magnetic, optical, or other physical device or means that can contain or store a computer program for use by or in connection with a computer-related system, method, process, or procedure. Programs can be embodied in a computer-readable medium for use by or in connection with an instruction execution system, device, component, element, or apparatus, such as a system based on a computer or processor, or other system that can fetch instructions from an instruction memory or storage of any appropriate type. A computer-readable medium can be any structure, device, component, product, or other means that can store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device.

[0037] The block diagrams and flow charts shown herein depict process steps or blocks that may represent modules, segments, or portions of code that include one or more executable instructions for implementing specific logical functions or steps in the process. Although the particular examples illustrate specific process steps or acts, many alternative implementations are possible and commonly made by simple design choice. Acts and steps may be executed in different order from the specific description herein, based on considerations of function, purpose, conformance to standard, legacy structure, and the like.

[0038] Referring to **FIGURE 2A**, a schematic graphical diagram illustrates an embodiment of a Real Time Supply Chain (RTSC) **200** that can be implemented using the ZLE architecture. RTSC **200** enables definition and implementation of best practices **202** that enable systems designers and users to effectively deal with competitive realities, adapt quickly to changes in operating and business environments, maintain competitive

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 181 of 302

advantage in an environment of technology globalization, reduce infrastructure costs, and create a proactive adaptive enterprise. The RTSC **200** includes components supporting supply chain visibility **204**, supply chain planning **206**, and supply chain analytics **208**, as well as other applications **210**. The supply chain components enhance operations and decision-making through real-time applications in applications that focus on using data to solve business problems.

[0039] The Real Time Supply Chain (RTSC) is being developed by Hewlett-Packard Company of Palo Alto, California, and defines a Foundation Layer **212**, which supplies multiple data services in combination with a starting data model **214**. The RTSC Foundation Layer **212** captures data from multiple SAP instances into a highly scalable centralized data store on a Hewlett-Packard Company NonStop Server, facilitating efficient movement of SAP documents in and out of a database. An RTSC data store **216** supplies global visibility to a corporation's data assets in disparate SAP instances. The starting data model **214** can be extended to meet particular functionality according to needs and desires of a particular customer. The Foundation Layer **212** supplies a robust starting data model for a manufacturing Supply Chain.

[0040] RTSC Foundation Layer **212** supplies a single view of a supply chain, enabling access to a single data source for Supply Chain applications **218**, for example including Supply Chain data management that collects, cleanses, enriches, caches, unifies, synchronizes, distributes data, and the like.

[0041] Referring to **FIGURE 2B**, a schematic graphical diagram illustrates an embodiment of a Real Time Supply Chain (RTSC) **200** including Supply Chain Data Management Components **220** that manage data in a data store **222**. The data management components **220** supply an infrastructure that maintains data fidelity, enabling customer applications to operate on complete and correct data. In various embodiments, supported customer applications may include SAP XI Adapters, Web Methods Adapters, and Hewlett-Packard Data Models. Similarly, data mapping, database updates, and subscriptions can be supported. The real time information director (RTID) is generally implemented as a data management component. Some systems may implement a customer/part deduplicator.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 182 of 302

[0042] In a particular embodiment, the RTSC Foundation Layer **212** is an application of Hewlett-Packard's Zero Latency Enterprise (ZLE) architecture to the Supply Chain business domain. The ZLE enables an enterprise to eliminate the latency in the ability to act upon information that is imposed when the desired information is contained within storage structures otherwise constrained to access by a specific application. Custom Foundation Layers can be specially-constructed for industries and business domains other than the Supply Chain. The individual Foundation Layers generally include a starting data model that is subject to customization. The individual Foundation Layers facilitate creation of customer-specific ZLE solutions.

[0043] The Foundation Layer **212** enables rapid customization and extension of the data model with no reprogramming, including the addition of more documents and tables. The Foundation Layer **212** also enables interoperation with multiple EAI vendors to attain enhanced or maximum connectivity to multiple SAP instances and other applications while supporting widely-accepted industry standards. In a particular embodiment, the Foundation Layer **212** supports Java language for application development and deployment and eXtended Markup Language (XML) for data input and output.

[0044] Referring to **FIGUREs 1A** and **1B** in conjunction with **FIGUREs 2A** and **2B**, the ZLE system collects data from various applications into a unified ZLE data store **104**. The ZLE data store **104** is an extended form of an operational data store, which contains current and historical data, including state data and event data. The data store **104** can be configured as massively scalable and highly available. A data handling system **100** using a Hewlett-Packard's NonStop Server™ using NonStop SQL/MX™ is well suited to serve as the platform for the ZLE data store **104**. Some application categories that can exploit the information in the ZLE data store **104** include business visibility applications, business activity monitors, business intelligence applications, campaign management applications, and customer relationship management applications. Some applications capable of supplying data to the ZLE data store **104** include enterprise resource management systems, payments systems, core business applications, and customer relationship management systems.

[0045] The illustrative embodiments enable usage of polymorphic metadata to create a ZLE solution that is customized to a particular enterprise environment. Various aspects of a data handling system **100** that are facilitated by polymorphic metadata include construction of a data model for the ZLE data store **104**, creating the feeds from data handling applications, mapping data from the data handling applications to the ZLE data store **104**, creating feeds to the consuming applications, identifying data to be supplied to the consuming applications, and defining when the data is to be delivered. Because the combination of applications, as well as application usage, is never exactly the same between different enterprises, tools are made available to enable rapid customization.

[0046] Customization of metadata is desirable for several reasons. Data models are often used differently, in the context of different companies and even within subsidiaries of a single enterprise. In a particular example, Systems, Applications, and Products (SAP) elements are used to define a data model. Some business-critical data is stored in customer-defined SAP elements. For a particular data handling system, modifications to predefined mappings are often desirable for each customer. For a single customer, several mappings are most likely highly useful to reconcile different uses of SAP within regions, subsidiaries, or acquired companies.

[0047] Customization is also useful due to the enormous number of SAP IDocs. Each customer is likely to promulgate more IDocs to meet objectives of a deployment. Often customers use extensions to the data model since a general data model typically does not capture all aspects of the SAP documents. Furthermore, subscribing applications may use information collected and aggregated from various SAP IDocs.

[0048] Referring to **FIGUREs 3A** and **3B**, schematic block diagrams illustrate an embodiment of a data services handler **300** that comprises a real time information director (RTID) **302** that maps documents into a data model. The data services handler **300** further comprises a disassembly logic **304** that disassembles documents for loading into a database **306**, assembly logic **308** that assembles documents from records in the database **306**, and a generator logic **310** that generates an eXtended Markup Language (XML) Schema that describes format of the documents. The disassembly **304**, assembly **308**, and generator logic **310** can operate using a single descriptive polymorphic metadata.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 184 of 302

[0049] In the illustrative embodiment, the data services handler **300** includes an Integration Server **312** such as SAP Netweaver XI, WebMethods, TIBCO or SeeBeyond and generally runs on an operating system such as HP-UX from Hewlett-Packard, Windows Server, or Linux. In various embodiments, the real time information director **302** runs under NonStop WebLogic Server (WLS) on the NonStop Server.

[0050] The Integration Server **312** supplies adapters **314** to obtain Interchange Documents (IDocs) **316** from SAP **318**. The adapter **314** converts the IDoc **316** into XML format if the SAP instance is unable to generate the IDoc in XML.

[0051] Documents are delivered to the NonStop Server **320** via JMS **322**, using a WLS JMS provider **324** in combination with a WLS client library **326** on the Integration Server **312**. The individual Integration Servers **312** supply a JMS adapter **328**. JMS is a widely adopted standard.

[0052] Individual SAP instances **330** are configured to generate SAP IDocs **316** upon the occurrence of defined significant events **332**. In the illustrative example, an order is entered or updated **332** in one of the SAP instances **330**.

[0053] The Integration Server **312** routes the document from SAP **318** to the NonStop Server **320**.

[0054] The real time information director (RTID) **302** supplies a Message-Driven Bean (MDB) **334** to acquire inbound IDocs from JMS. Written in the Java Programming Language, a bean such as an enterprise bean is a server-side component that encapsulates business logic of an application. The business logic is code that performs the application. A message-driven bean is an enterprise bean enabling J2EE applications to process messages asynchronously. The real time information director **302** disassembles the IDocs **316** into various records and inserts them into the RTSC data store **306** using prepared SQL statements. Document metadata defines how the elements of the XML IDoc maps to rows and columns of the data store. A different document metadata is generally used for each type of IDocs **316**. In the illustrative example, the Disassembler **304** uses the metadata for an ORDERS05 IDoc **316**.

[0055] The real time information director **302** receives documents from Systems, Applications, and Products (SAP) called Interchange Documents (IDocs). The real time information director **302** can publish the IDocs via subscriptions that are expressed in eXtended Markup Language (XML). The real time information director maps the IDocs to and from a data model, for example Hewlett-Packard's Real-Time Supply Chain (RTSC) data model. The real time information director **302** uses metadata to do the mapping. Flexible functionality is attained for the real time information director **302** via usage of easily customizable metadata.

[0056] Referring to **FIGURE 3B**, many applications receive documents from the RTSC data store **306** using a publish-subscribe paradigm. The subscription is configured in the real time information director **302**. The subscription definition includes a triggering event, the document to be generated, and the name of a JMS queue or topic to which the document should be delivered. Applications directly receive the published document by reading the JMS queue or topic.

[0057] The Integration Server **312** can read the JMS queue or topic and route the document to various applications, including other SAP instances **330**.

[0058] The subscription can be implemented using a "select from stream" feature of NonStop SQL/MX **338** that receives data in a stream. The event is defined in an SQL statement, so the user has all the power of the SQL language to define events narrowly or broadly. The number of subscriptions that can be configured by Subscription Metadata **342** has no fixed limit. A single document inserted or updated in the database can generate many subscriptions, or one or none.

[0059] Content and format of the document generated is defined using Document Metadata **340** and may be very different from the incoming document that triggered the subscription. However, the same document metadata may be used for both inbound and outbound documents. Separate maps need not be created for inbound and outbound documents that have the same content and format.

[0060] The real time information director (RTID) **302** assembles the document in accordance with the Document Metadata **340**, gathering records from various tables related to the triggering event. The document is output in XML and placed on the JMS queue or topic. The output document may be an IDoc, or it may be a "native" document. The native format enables the real time information director **302** to publish documents that contain more information than is allowed within any specific IDoc. Native format documents are also easier to define, as only document content metadata is used and format metadata superfluous. The real time information director **302** exports XML schema for native documents from the document content metadata, so that other applications can process native documents.

[0061] The real time information director **302** is a set of data services that forms a core of the Foundation Layer. The real time information director **302** serves as the gateway between the ZLE data store **306** and both supplying and consuming applications **336**. The real time information director **302** maps the incoming data to the data model and delivers requested data to consuming applications, either on demand or by subscription. In this process, the real time information director **302** cleanses and enriches data, provides data security, and insures data integrity.

[0062] The Foundation Layer includes a starting Supply Chain data model that further has mappings for SAP IDocs **316** that are core to any supply chain (Orders, Invoices, Shipments, Inventory, Suppliers, Bills of Materials, and the like). The Foundation Layer also includes real time information director **302** software that is used to efficiently move SAP IDocs **316** in and out of the database. The Foundation Layer also implements instructions for extending and customizing the data model to additional IDocs **316** and variations of the documents, as well as instructions for configuring the RTSC environment, including connections to an EAI tool to access SAP instances **330** (SAP Netweaver and WebMethods Integration Platform).

[0063] Typically the integrity and security of data is the responsibility of a single unified application that owns the data in a model works well with enterprise applications. However, few large enterprises are built around a single enterprise application. Many large corporations have hundreds of enterprise applications. The ZLE system enables and facilitates data sharing for enterprises that run a large number of applications. In the ZLE

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 187 of 302

system, the real time information director **302** operates as the application that guarantees the integrity and security of the ZLE data store **306**. Operations of the real time information director **302** are directed by polymorphic metadata, metadata that enables the rapid customization of the ZLE data model and integration of multiple various enterprise applications with the ZLE data store **306**. Polymorphic metadata defines the security model and data integrity rules, and defines the mapping between external document formats and the data model.

[0064] Functionality of the real time information director **302** can support specific criteria of particular industries and application frameworks, a flexibility that is assisted by a capability to create additional and novel classes of metadata.

[0065] In a particular embodiment, the real time information director **302** can be implemented using the Java™ Language, for example supported under BEA WebLogic Server for HP NonStop Servers (WLS). In other embodiments, the real time information director can be implemented in other languages and/or used in conjunction with other application servers. The illustrative real time information director **302** can be compliant with J2EE standards. The real time information director **302** can use the application server for connectivity and interoperability with other products and platforms, especially through JMS (Java Messaging System) via the WLS JMS provider, and HTTP via Servlets. The real time information director **302** can be managed through the same interfaces as other WLS components and can implements a JMX Bean to supply metrics and enable online deployment of new metadata. The real time information director **302** may use a standard WLS logging facility to report events and exceptions. Any management platform that can manage WLS, such on Hewlett-Packard's OpenView, can manage the real time information director **302**. The WLS environment can be further leveraged to publish real time information director **302** interfaces as Web Services.

[0066] Real time information director components can also be used outside of an application server, using file inputs and outputs, as part of a batch application running directly in a Java Virtual Machine. Accordingly, the illustrative real time information director can be implemented for usage with other application servers or with no application server at all.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 188 of 302

[0067] The Foundation Layer can be used to build a centralized RTSC Data Store that collects information from across the enterprise, an application that is facilitated by a highly scalable, available and reliable platform, as is provided by HP NonStop Server. For the most part, the platform is transparent to the real time information director **302**. However, the real time information director **302** can use of features specific to NonStop SQL/MX, which is only available on the NonStop Server. For example, the Subscriptions Service of the real time information director **302** uses features unique to SQL/MX.

[0068] Data is travels to and from the real time information director (RTID) **302** in the form of XML documents. The RTSC Foundation Layer is designed to handle SAP IDocs (Interchange Documents) expressed in XML, and is compliant with the XML schemas published by SAP. Other Foundation Layers, for example for usage in other business problem domains, may support other widely adopted XML document types. Logic in the real time information director **302** to support SAP IDocs can be extended to other applications of XML. The real time information director **302** also receives and publishes XML documents in native format, which closely reflects the structure of the data in the RTSC or ZLE data store **306**. The real time information director **302** exports XML schema for native format documents, which other applications can use to more easily interoperate with the Foundation Layer. Two basic services offered by the real time information director **302** include disassembly of inbound XML documents into multiple records in preparation for insertion into various tables, and assembly of outbound XML documents from multiple records selected from various tables. The services can be driven entirely by polymorphic metadata.

[0069] In an illustrative embodiment, the real time information director **302** can contains no procedural code that is hardwired to a specific data model. Instead the program logic of the real time information director **302** can consume descriptive metadata to identify individual specific data elements, and the behaviors, for example program logic, applicable to the individual elements.

[0070] Software code that is adapted to a customized data model, without using direct modification, is often called "table-driven code". Information about the data is itself stored in one or more 2-dimensional matrices or tables. Each row of a metadata table defines a data element, or related group of data elements, and each column defines a

specific attribute of those elements. Usage of tables and matrices enables definition of any particular data element without writing specific behaviors or program logic. Instead the desired behaviors can be inferred from one or more attributes of each element. Program logic can simply read the table to determine which behaviors are applicable to each data element. However, the table-driven approach does not handle new behaviors or special cases well. Each special case or new behavior that is not uniformly applied requires the introduction of new columns to the metadata, resulting in metadata tables that are sparsely populated and difficult to maintain. Common patterns are not easily managed. Breaking down the metadata table into various sub-tables can ameliorate these problems, at the cost of additional complexity.

[0071] In the illustrative embodiment, the real time information director **302** does not rely principally on 2-dimensional tables to represent metadata. Instead the real time information director **302** represents metadata as polymorphic objects. Polymorphism enables different metadata objects to have different numbers and types of attributes, determined by class. Behaviors can be defined that are specific to certain classes of objects, and the attributes specific to those behaviors need only be specified in those classes of objects, an approach that facilitates introduction of new behaviors and special cases, without modifying the metadata for non-applicable or irrelevant objects and classes.

[0072] The metadata objects for the real time information director **302** can be constructed in the Java programming language. Individual classes of metadata can be embodied in a distinct Java class. Special cases of metadata are frequently represented as subclasses of more general metadata classes. Complex metadata can be represented as composite objects constructed from several metadata classes.

[0073] In an illustrative system, metadata objects are stateless. The metadata objects are not modified after initial construction, a characteristic attained, for example, in the Java language by declaring all attributes of metadata classes as *final*. Such attributes are set only during the construction of the metadata object. The attributes are specified as parameters of the metadata object, or are constructed entirely from such parameters and other information available during the construction of the metadata object.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 190 of 302

[0074] Metadata classes do implement behaviors for the data described in the metadata class. The power of polymorphism resides in the ability to override methods of the metadata super classes. The methods can utilize the unique attributes specified in the constructor of the metadata subclass. However, since the metadata class is stateless, each method of the metadata class is passed at least one stateful object containing the data on which the method operates.

[0075] As a general rule in object-oriented design, one strives to embed stateful data in classes containing the methods that operate on the stateful data. In contrast, in the real time information director **302**, methods of stateful classes generally delegate operation to stateless metadata objects.

[0076] Metadata objects are far less numerous, are subject to much less frequent modification, and are much more frequently used, than individual data objects. Significant performance advantages can be attained by caching metadata objects for reuse. Performance can also be improved by sharing metadata objects between threads. Optimizations can be attained when the metadata objects are stateless. When metadata is modified, obsolete metadata can be discarded and new metadata instantiated.

[0077] Overridden methods of a metadata class represent a behavior that is unique in some regard, and that generally applies only to certain data elements. The potential always exists that other data elements, identified in the future, may use a similar behavior.

[0078] Consequently, the identification of specific data elements is generally not hard-coded within the implementation of the class. Instead the identification of a specific data element can be specified in a parameter of the constructor of the class, or be identified indirectly.

[0079] Specific data elements to which metadata apply are specified in parameters of the constructors for various metadata classes. The constructors can be nested to represent hierarchical relationships between described data elements. Identifying information about the hierarchical context for a metadata element can be derived from the containing metadata object.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 191 of 302

[0080] In the real time information director (RTID) **302**, contained metadata object is usually constructed before the containing metadata object, and reference to the contained object is passed to the containing object. The immediate result of the nested constructors is a tree with all references pointing from the root, through intervening branches, to the individual leaves. In some cases, a more complex interconnection of the metadata may be used, for more convenient and efficient utilization. For example, certain branches might use a reference back to the root, to the parent branch, or to a branch in a parallel tree. In such cases, metadata classes implement methods to traverse the tree. Such methods are passed references to the objects to be referenced within the tree, and which may be traversed in parallel. The methods are typically called only after the construction is complete, and consequently the references being updated thereby cannot be declared as *final*. To insure class statelessness, traversal methods can generate an exception if reinvoked with different references.

[0081] Nested constructors can easily and conveniently be invoked directly in a Java source file. Java source code is a concise and flexible expression of customized metadata for the real time information director **302**. For example, customized metadata for each SAP IDoc resides in a distinct Java class. The source file of the Java class has one line for each tag in the XML document that is pertinent. Each line contains a *new* statement to instantiate a metadata class corresponding to the tag. A different metadata subclass exists for each style of mapping. The name of the tag is one parameter of the constructor. Usually another parameter is the name of a table or field in the database. If the data element represented by that XML tag contains nested elements, then another parameter of the constructor is an array of definitions for the tag. The array is constructed with a new statement, and populated immediately by new constructors for the various contained data elements. An example is as follows:

```
new Segment("E1EDK01", new FieldElement[] {
    new FieldElement("ACTION", "SALES_ORDER_ACTION_CD"),
    new FieldElement("CURCY", "CURRENCY_CD"),
    new FieldElement("WKURS", "PRICING_STATISTICS_EXCHANGE_RT"),
    new FieldElement("ZTERM", "PAYMENT_TERMS_CD")
    }),
```

[0082] Metadata constructors need not be directly nested. A metadata object can be separately named and constructed, and then referenced within a subsequent constructor. For example, a metadata object can be referenced in several other metadata objects. In a particular example, the metadata object that defines an IDoc header can be constructed once and referenced in the metadata classes for many different IDocs.

[0083] Metadata can be formed in a form other than Java source code. A graphical tool can be built to edit the metadata. One risk in usage of a graphical tool is that the flexibility enabled within Java may be lost. A metadata representation should be defined such that polymorphism of the metadata is not unduly restricted, a risk that is more likely when using a graphical tool.

[0084] Java classes that purely define metadata contain constructor invocations alone. The only statements present are "new" statements. The parameters of the new statements are either identifying or descriptive attributes of data elements. References to contained data elements and references to other metadata objects that identify peculiar behaviors applicable to the contained data elements.

[0085] Formatters are an example of the other metadata objects. Formatter is an abstract class that can be passed as a parameter to elements of the XML mapping metadata for individual fields. The Formatter defines transformations between the elements as represented in the SAP IDoc and as represented in the database. Each concrete Formatter class defines a particular kind of modification to the format of a data element. Some, but not all, formatters take parameters in the constructor, such as a length for a padding formatter. The same Formatter can be used for many different fields and thus is often constructed separately.

[0086] Customized metadata is descriptive. The parameters of the constructors identify *what* is to be done, not *how* the operations are performed. *How* is defined by methods of the metadata classes.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 193 of 302

[0087] The same customized metadata can be used for multiple purposes. The XML document mapping metadata is currently used in three different ways. First, the metadata is used to disassemble the elements of the inbound XML document into rows and columns to be inserted into various tables in the database. Second, document-mapping metadata is used to select rows and columns from various tables of the database and assemble the elements into an outbound XML document. Third, the metadata is used to generate an XML schema for the documents that can be input or output using this metadata. Even the Formatter classes have the ability to transform in both directions and generate XML schema.

[0088] Different uses of the metadata can be embodied in different methods of the metadata classes. Each of the different methods can be overridden by subclasses of the metadata. Individual data elements described by these metadata classes may be defined a single time and still realize all different uses.

[0089] Polymorphic metadata has functional characteristics that improve over other techniques. For example, most data transformation tools generate maps or style sheets that transform in one direction only. Reverse transformation uses a separate map or style sheet, a deficiency that arises since the mappings are, at least in part, procedural rather than descriptive. In a specific instance, the other data transformation tools allow expressions and calls to various transformation functions. XSLT (eXtensible Stylesheet Language Transformations) Style Sheets are often used to transform XML documents. XSLT is a very high level scripting language, implemented in XML, which enables implementation of a tree-oriented transformation language for transmuting instances of XML. Comparable style sheets can be constructed that disassemble SAP IDocs into rows and columns in the database. XSLT is a procedural language and thus can be used to implement all mappings directly. However, resulting style sheets for the IDoc are highly verbose, measured in terms of either characters or lines, in comparison to the corresponding Java metadata class for the IDoc. The resulting style sheets transform in only one direction so that a separate style sheet is implemented for the reverse transformation. Also the XSLT style sheet is an order of magnitude slower to execute than the real time information director mapping.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 194 of 302

[0090] When a new pattern of mapping or reformatting is to be implemented, a new Metadata class and relevant methods are constructed in Java to represent the new pattern. Definition of the pattern and methods often expends a higher initial investment than expressing the new mapping using the high level language of XSLT or the expressions and functions built into a data transformation tool. However, the initial investment is recouped as other elements can be mapped using the same new metadata classes concisely and in both directions.

[0091] Many different systems allow extension through mechanisms such as user-exits or call-backs. Such extensions are usually cumbersome to implement and integrate into the system. Identification of potentially useful user-exit points in the system, providing full access to the data in the parameters of user-exits, and leveraging of other mechanisms of the system within user-exits can be difficult. Polymorphic metadata enables a much more powerful and flexible approach to extending the capabilities of the real time information director **302**. Any non-protected, non-final method of a metadata class can be overridden in a user-written metadata subclass, a mechanism that enables a narrowly targeted callback and enables full access to non-protected attributes and methods of the super class.

[0092] In the illustrative embodiment, document metadata is the primary metadata on which the real time information director **302** resides. Document metadata defines the content and format of documents received and published by the real time information director **302**. The real time information director **302** is customized and extended either by modification of the metadata for an existing document type, or by the creation of a new document type. A top XML tag identifies the type of a document. Document metadata is defined once for a document type, whether the document is designed for inbound use, outbound use, or both inbound and outbound use.

[0093] Document metadata is subdivided into two parts. The first part defines the *content* of the XML document. The second part defines the *format* of the XML document. For SAP IDocs or for any other externally standardized document type, both parts of the document metadata are explicitly defined. For Native Format documents, the format is automatically determined from the schema of the tables identified in the content metadata; therefore no format metadata need be specified.

[0094] Common document content metadata identifies tables that supply data to or receive data from the document, and association criteria for grouping related records within the hierarchy of the XML document.

[0095] The real time information director **302** automatically generates and caches prepared SQL statements from the content metadata after reading the schema of the tables identified in the metadata to obtain detailed information about names and data-types of individual columns. Parameterized select, insert, delete, and update statements are generated according to application to enable the real time information director **302** to manage inbound and outbound documents.

[0096] Additional document content metadata may be supplied for outbound documents. Summary metadata identifies how detailed records can be aggregated for documents containing summary data. Summary metadata causes SQL statements to be generated using the GROUP BY clause.

[0097] In most cases, document content metadata does not have to give information on individual fields, since the information can be obtained from the database schema. Document format metadata, in contrast, contains at least one line of metadata for each column populated by the document. Consequently, the quantity of metadata used for a native format document is a small fraction of that used for an externally defined format, such as an IDoc.

[0098] One benefit of separating content metadata from format metadata is a capability to avoid specifying format metadata for native documents. Another benefit is metadata simplification. For example, selected columns of several tables can be populated by the same element of the inbound document. Specifically, the order number can appear in every record associated with an order IDoc. The mapping of the appropriate tag (<BELNR>) to the appropriate column-name (SALES_ORDER_NO) appears only once in the metadata. From the content metadata and the schema of the database can be determined all rows that have a column by the name. In contrast, in the procedural XLST style sheet, the mapping is repeated for each record, enabling metadata to be less redundant and thus less verbose.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 196 of 302

[0099] A further consequence of the separation of the format and content metadata is a capability to modify the database schema without modifying the metadata. For example, fields may be inherited from parent or grandparent elements of the document to speed retrieval of information for later usage. Such an operation is termed *denormalization* in database applications and is used to eliminate the operation of joining detailed records in database with the parent records. A decision to denormalize data can be done with no change to the document metadata. Also columns can be dropped from a table without change to the document metadata. A document element can be mapped to a column-name that does not currently exist in any table.

[00100] The real time information director **302** enables multiple document types to be defined that share the same content metadata, but which have different format metadata, a condition that frequently occurs when integrating multiple SAP instances with the ZLE system. Each SAP instance supplies the same basic sets of documents including orders, invoices, shipments, and the like, but mappings for each set has variations, especially for customized elements. A separate document definition for *orders*, as an example, can be created for each source, containing only format metadata and referencing common content metadata. Usage of the document sets eliminates redundancy and also enables usage of the same set of cached SQL statements to load orders into the database, regardless of which source system sent the data.

[00101] The top-level tag of the inbound XML document is typically used to identify which document definition is used. Often, the appropriate document definition to use is unknown until the IDOC header is parsed to identify the source system. The real time information director **302** enables switching of the document metadata class after parsing the header.

[0100] Inbound documents can be delivered to the real time information director **302** synchronously, using HTTP, or asynchronously, using JMS. A servlet can be supplied that invokes the real time information director **302** via an Enterprise Java Bean (EJB) whenever documents are delivered via HTTP. The real time information director **302** is invoked via a Message Driven Bean (MDB) whenever documents are delivered via JMS. In either case, the Bean is stateless, and serves only as a thin layer to access the services of the real time information director **302**.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 197 of 302

[0101] Outbound documents can be requested from the real time information director **302** synchronously, using HTTP. A servlet can be supplied that invokes the real time information director **302** via an EJB whenever a document is requested via HTTP. Outbound documents can also be delivered asynchronously, using a JMS subscription.

[0102] Referring to **FIGURE 4**, a schematic block diagram illustrates an embodiment of a data handling system **400** in a configuration suitable for handling subscriptions to documents. The real time information director **402** enables other applications to subscribe to documents. Subscriptions are triggered by insertions or updates of data in the RTSC data store **406**. Subscriptions can be extremely flexible, for example a subscription can be defined that is triggered by any change to a table, by changes to specified rows of the table, or by specified types of changes. For example, one subscription may be triggered by orders from a specific customer, while another subscription is triggered whenever an order is canceled. No limit is typically imposed on the number of subscriptions that can be created, or the number that can be triggered by a single event.

[0103] The definition of a subscription includes specification of three parts: (1) a triggering event, (2) a type of document that is generated, and (3) a JMS queue or topic to which the document is delivered. The triggering event can be specified as an SQL select statement. The type of document is a reference to document metadata **404**, enabling definition of both the content and the format of the document. The subscription capability, in combination with document metadata, enables the real time information director **402** to supply any data to an application as rapidly as possible.

[0104] The real time information director (RTID) **402** may include a rules engine **410**, a scoring engine **412**, and an aggregation engine **414** to enable rules and scoring to be used in the context of a subscription, for exception handling. For example a subscription may be triggered by any order with a large quantity. Real-time scoring may detect the likelihood that the order is entered incorrectly or by someone who is defrauding or gaming the system. Business rules **420** can place orders on a queue to enable checking and correction by service agent.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 198 of 302

[0105] The real time information director (RTID) **402** further includes the aggregation engine **414** to compute aggregates in memory from data selected from the database **406**. The aggregation engine **414** is primarily supplied for client applications and services that extensively use aggregated data, such as real-time scoring and business rules. Such applications use a variety of aggregates, based on different conditions, which generally cannot be computed in a single SQL statement.

[0106] Some aggregated data is accessed frequently. For example, an aggregate may be continuously updated on a business activity monitor. Or an aggregate can be frequently utilized in rules or mining models. If the cost of the creating the aggregate from raw data is also high, efficiency may be served by maintaining a persistent aggregate. A persistent aggregate may be implemented by creating a state table in the data store **406** to hold aggregates relevant to a particular item. For example, a state table may contain supply and demand totals by product, region and/or date. The table is continuously updated as the orders and supply documents flow through the system.

[0107] Document metadata **404** can be maintained to include accumulator definitions. An accumulator defines the relationship between an element of an inbound document and an element in a state table. An accumulator can represent a total, a count, a maximum or a minimum. The accumulator updates the value of the element in the state table, based upon the value of the element in the document, after taking into account the prior value of that element if the document replaces an existing document.

[0108] Outbound documents can directly reference the rows of the state tables to deliver the aggregates directly. Also, aggregate definitions can reference the state tables to speed computation of complex aggregates.

[0109] A SQL Select statement has a single *WHERE* clause that applies to all aggregates computed by that statement. In contrast, each aggregate used by business rules or scoring models may have different conditions, calling for separate computation. SQL aggregation can have a complementary role to the aggregation engine **414**. Document metadata **404** can direct SQL to total data into buckets grouped by all the dimensions of the data relevant to the conditions of the aggregates, a practice that reduces the number of rows to be aggregated in memory by the aggregation engine **414**. Aggregates are defined

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 199 of 302

separately from document metadata **404**, using context-sensitive graphical wizards. The wizards read the document content metadata and database schema to expose the records and columns that are available to be aggregated. The wizards present the appropriate aggregate functions applicable to the data types of the elements. A separate set of aggregates can be defined for each document type. Because business rules and scoring models frequently change, intelligent wizards can be used to quickly modify and redeploy aggregate definitions.

[0110] The aggregation engine **414** supports both simple aggregates, such as sum, min, max, mean, first, last, count and count-distinct, and compound aggregates such as ratios. Ratios are frequently used in data mining. Examples are weekend calls to weekday calls, evening calls to daytime calls, and long-distance calls to local calls. Aggregates can also utilize complex conditions.

[0111] The aggregation engine **414** supports the creation of custom aggregate functions, both simple and compound, user-written conversion methods, and user-written methods to compute derived attributes. User-written functions and methods are exposed to the aggregate wizards through Java reflection.

[0112] In some cases, a customer may not wish to license a third party product such as Blaze or Trillium to perform certain data enrichment or data cleansing tasks. The customer may wish to implement such enrichments or cleansings in Java code. The functions are encapsulated in a Java class that subclasses the real time information director's Enrichment class. The user may specify such user written Enrichment classes in the document content metadata.

[0113] Many of the services, such as data cleansing, aggregation, real-time scoring, business rules, and the like are invoked from the document content metadata as special enrichment classes that are provided ready-to-use with the real time information director.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 200 of 302

[0114] The rules engine **410** supports specification of business rules without writing Java code. In some embodiments, the rules engine can be implemented using Blaze Advisor, from Fair-Isaac. The scoring engine **412** enables real-time scoring based on data-mining models. In an illustrative embodiment, the mining models are built using SAS Enterprise Miner, and deployed using Genus Mining Integrator. The aggregation engine **414** computes derived values and aggregates. The aggregates are used in combination with the scoring engine **412** to add value to the business rules. The services are invoked via document metadata **404** and supply additional metadata.

[0115] Real-time scoring is a useful capability of Hewlett-Packard's Zero Latency Enterprise architecture and is discussed in greater detail in United States Patent Application Serial Number 10/423,678, and corresponding provisional application number 60/383,367, entitled *Zero Latency Enterprise Analytic Learning Cycle* which is incorporated by reference in its entirety. The real-time scoring capability has previously been enabled under a NonStop CORBA framework and is now available as part of the real time information director **402**. Scoring is the penultimate step in the application of data mining. The steps include, first identifying a behavior to be predicted to enable more intelligent and proactive actions. For example, fraudulent behavior may be prevented if predictable. Similarly, prediction of a customer's likelihood of buying various products can result in presentation of better offers. A second step in real-time scoring is collection of past data that reflects known outcomes, both positive and negative. A third step is profiling of data to identify attributes that are possible predictors of an outcome. Each predictor attribute is aggregated from detailed raw data. The number of different stores visited by a customer during the past month is an example of an aggregate.

[0116] A fourth step is transforming, aggregating, and downloading data into a data-mining tool, such as SAS Enterprise Miner. Fifth, a predictor model can be built using tools in the data-mining products. Step three can be repeated to generate different aggregates, based on insights gained from the tools. A sixth step includes usage of the mining model to score all the entities, for example customers, for which the outcome is not known. The score is a probability, typically a value between 0 and 1, of the behavior being modeled. A seventh step is usage of the score in the next opportune business decision.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 201 of 302

[0117] One benefit of the ZLE application of Data Mining is that the time gap between the two final steps is eliminated. Usually scoring is performed as a batch process. Significant time may elapse between the time that the batch process is completed and the time that a customer walks into the store again. As a result, significant activity may have occurred since the score was generated that makes the score obsolete. The scoring engine **412** in the real time information director **402** utilizes aggregates generated from current data and delivers the score to business rules for immediate action. The process is cyclical. As knowledge gained from data mining is used, new insights are available that lead to improved models. Also, mining is performed based on outcomes of decisions are made from application of earlier models. The process involves frequent updating of aggregate definitions. The agility enabled by the aggregation engine **414** and Aggregate Wizards is therefore highly useful.

[0118] The Foundation Layer interoperates with Genus Mining Integrator for NonStop SQL. The Genus Data Preparation Tool performs the profiling step and data transformation steps on the NonStop Server. The Genus Data Transfer Tool transfers the data from the NonStop Server to Enterprise Miner. After the prediction models are developed in Enterprise Miner, the Genus Model Deployment Tool deploys the completed models back to the NonStop Server where the Aggregate Wizards and Scoring Engine can access the models.

[0119] A subscribing application can manage exceptions. For example, a subscribing application may subscribe to specified high-value events, for example placing or a large order, a disruption in supply, and the like.

[0120] A subscribed document may be a document other than an IDoc. The document can contain all information related to that high-value event. Any miscellaneous information for usage by the subscribing application for addressing the event may also be contained in a document, for example inventory, projected supply and orders of a particular product.

[0121] A subscribing application can read XML documents from a JMS queue, provided that an XML schema is available.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 202 of 302

[0122] The real time information director **402** can be used to implement various additional data services. Document metadata can be used to drive invocation of the services.

[0123] Referring to **FIGUREs 5A** and **5B**, schematic block diagrams illustrate an embodiment of a data handling system **500** that implements security and auditing features. Security and auditing operations are defined according to two persons, or pseudo-persons, who are identified in the header of a document sent to the real time information director **502**. One person is identified as an *agent*, a person who requests information in a query document or submits changes to the database. The other person is identified as a *consumer*, a person whose information is requested or updated.

[0124] In an alternative setting, for example a healthcare setting, the consumer is typically a *patient*, and the agent is typically a healthcare provider, administrator or insurance agent. In a finance setting, the consumer is typically a *customer*, and the agent is typically a loan officer, teller, or customer service agent. In the hospitality industry, the consumer is typically a *guest*. Terms agent and consumer are selected as industry-neutral.

[0125] A secure or audited document has a header that includes an external identifier (ID) of the agent and the external ID of the consumer. An external ID typically has two elements, each in the form of a character string: an external ID proper and a name or code for an issuing agency. For example, the external ID may be "123-45-6789", or "SocialSecurity".

[0126] The real time information director (RTID) **502** resolves each external identifier (ID), agent, or consumer to a unique internal ID by reading a *Person* table of the database. In one example, the internal IDs are 64-bit numbers assigned by the real time information director **502**, using a "Key Manager" functionality, the first time that a consumer or agent is registered in the system. Documents that register new users generally use a SystemKey enrichment.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 203 of 302

[0127] Multiple external IDs may be associated with the same person, issued by different agencies. For example, a person may be identified using a blue cross number, a social security number, an account number, and a driver's license number. Before registering a person under a new external ID, a determination is made of whether that person already is registered under another external ID. For the example of an Electronic Health Record (HER), a query can be made to look up persons with the same last name, date of birth, and zip code. If a matching person is found, an additional external ID can be assigned to the same internal ID.

[0128] Identities of agents and consumers can be managed in the same way to enable consumers to directly log into the system using the same external ID normally used when contacting a business. In such self-service transactions, the external ID of the agent and the consumer are the same.

[0129] Not all documents are secure or audited. Non-secure and non-audited documents may omit a header. The document definition indicates whether the document is secure and/or audited.

[0130] Not all audited or secure documents refer to one specific consumer. The header of a document may omit a consumer if the information in the document does not apply to a specific consumer. For example, a query that returns a list of patients for a document and the registration document for a new patient generally does not contain a consumer ID. However, an agent is generally always included in an audited or secure document.

[0131] The real time information director **502** is generally not used for authentication. For example, authentication may be performed by a portal **512** before sending transactions to the real time information director **502**. The real time information director **502** typically accepts information from the portal **512** without additional verification. The connection from the portal **512** to the real time information director **502** is secured through techniques transparent to the real time information director **502**.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 204 of 302

[0132] Audited documents cause a record to be written to an *AuditHeader* table. The AuditHeader contains the internal ID of the agent and consumer, the date/time at which the transaction occurred, as well as an event date/time specified by the sending system, and the name of the document received. Each AuditHeader record gets a unique ID assigned by the Key Manager called the *AuditId*, a primary key of the AuditHeader. The AuditHeader also has an index enabling efficient lookup by date and consumer. Auditing is implemented primarily for the benefit of the consumer and has a primary intended usage to protect privacy rights of the consumer and protect against malpractice.

[0133] A DocumentDefinition for audited documents specifies additional details for recording in combination with the header. Details are stored in an AuditDetails table and are generally expressed in XML. In one example, the details can include two types: an inbound document received by the real time information director **502** and an outbound document sent by the real time information director **502**. The two detail types apply to both queries and updates. Other types of details may be implemented, for example that apply only to updates. Two possible detail types are an image of corresponding data in the database before update and a database image after upsert.

[0134] A primary key of an audit details record can include *AuditId*, the type of details such as Inbound or Outbound, and a sequence number. The Audit system can break up XML documents larger than 4K into smaller chunks stored in several records.

[0135] The real time information director **502** can support document definitions that query the audit table by consumer and date, and can also support *AuditDocumentDefinition*s, query documents that return one of the audit details *as* an XML response of the request.

[0136] The Audit system can also automatically store AuditId of the current transaction in any or all affected records in an update document. Typical data models call for the date/time and ID of the agent who last updated the record to be stored in various tables. In contrast, using AuditID saves space and enables much more information to be captured about the last update, for example including transaction type and possibly full details of a change if storage of such an information capacity is warranted. The real time information director **502** can also be enabled to separately record AuditId of the original

record insertion date since some data models carry both created-by/when and last-updated-by/when fields.

[0137] The Security system may also be configured primarily to guarantee consumer rights including privacy, for example relying on the portal **512** to supply first level authorization of agents in addition to authentication. The portal **512** may use a directory service to perform security functions. The directory may grant rights to perform certain transactions, for example submit corresponding documents. The directory generally cannot manage rights granted by the consumer. Rights granted by the consumers are stored in the Real Time Data Store and managed by the real time information director **502**. Consumer-granted rights determine *which* agents have access to *each* consumer's data.

[0138] Consumers may grant permission to agents through a variety of different models, called *Security Policies*. A simple but impractical security data model is a list for each consumer of all individual agents that have the right to the consumer's data. A more complex but also more practical model supports multiple security policies, each with an underlying data model. For example, the real time information director **502** can support creation of standard and custom policies. One possible standard policy is a self-service policy that grants a consumer, either patient or customer, access to their own data. A second standard policy is a personal agent policy that grants individually-named agents of individual consumer's access to that consumer's data. An example is a patient's primary care physician.

[0139] New or custom policies can simply be added. Example policies include a guardian policy that grants a consumer access to the data of a legal dependent, and an agent group policy that grants members of a named group access to a consumers data, for example members including doctors in a named Emergency Room.

[0140] Policies can be applied in any combination to individual transactions. For example, rights to perform a transaction A may be granted by either the self-service or guardian policy, while transaction B is granted by either the personal agent or agent group policy.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 206 of 302

[0141] A policy is generally defined in document metadata **504**, using the metadata to specify and customize policies. By definition, a document is secure if and only if a security policy is identified by name as one of the options in the document definition. Security policies are implemented in concrete subclasses of *Policy*. Like document definitions, concrete policy classes have a default constructor.

[0142] For example, a document option "Policy=SelfOrPersonalPhysician" causes the class *SelfOrPersonalPhysician* to be instantiated. The policy creates subclass *CompoundPolicy* and calls an associated super class with the string "SelfService PersonalPhysician," instantiating concrete policy classes *SelfService* and *PersonalPhysician*. CompoundPolicy enables the agent to access the consumer's data if any of the referenced component policy classes grant access. Class SelfService grants access if the consumer ID and agent ID specified in the document header match. PersonalPhysician creates subclass *PersonalAgent* and passes the agent relationship name "Physician" to the super class. Subclass PersonalAgent grants access if the *ConsumerAgentRelationships* table contains a record associating the agent to the consumer, for example as "patient" to "physician."

[0143] CompoundPolicy and PersonalAgent are examples of standard abstract policy classes provided as part of the real time information director **502**. Custom policies are typically created by subclassing such standard abstract policy classes, and referencing other concrete policy classes. PersonalPhysician and SelfOrPersonalPhysician represent the type of concrete policy classes implemented as part of a customization process for a particular industry or business. SelfService is a concrete Policy class supplied with the real time information director **502** that is applicable to any industry. Documentation is typically supplied that enables professional services to create additional abstract policies.

[0144] SelfService policy does not access the database but rather compares elements of the document header. Most policies access the database, generally by implementing a template method *constructContext*. The method is called each time a document handler is created that uses the policy. The method is passed a SQLConnection so that the Policy can create prepared statements. However, the connection and prepared statements do not belong to the Policy as metadata, which is instantiated only once, but to the document handler that likely is instantiated several times. Therefore, the Policy creates a clone and

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 207 of 302

returns the clone as a result of the constructContext method, a result achieved from the constructContext method by calling an alternate constructor of the same class, taking the clone and the connection as parameters. Statements are prepared in the clone constructor using data stored in the metadata class.

[0145] Concrete policies are instantiated by an internal class of the real time information director **502** called the PolicyManager. PolicyManager ensures each concrete policy metadata class is instantiated only once during any execution of the real time information director **502** no matter how often the class is referenced in the metadata. However, the policies can clone as part of DocumentHandlers by implementing the constructContext method.

[0146] Concrete classes similar to PersonalPhysician can be written without dealing with details of cloning. PersonalAgent automatically manages operations to clone the class and create the prepared statements. PersonalPhysician simply customizes behavior by specifying the value "Physician" for one of the fields in the query. Operations are transparent to the person writing the concrete class.

[0147] A small number of data models may be supplied to represent agent groups, consumer-group relationships, and consumer-guardian relationships, and the like. Several abstract policy classes can be created to use the different data models. Some enterprises may modify the abstract policy classes or create new policy classes to customize the underlying data models. Some enterprises may supply additional information about the relationships between agents and/or consumers in a dedicated data model.

[0148] The header may also include an element that identifies the particular role of an agent when a user logs onto the system. The role can be passed in the document header. The Policy may reference the role when applying policy. For example, the PersonalPhysican policy may grant access only if the role specified in the header is "Physician" and the SelfService policy may grant access only if the role specified in the header is "Self".

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 208 of 302

[0149] The real time information director (RTID) **502** is useful for building a secure ZLE environment. The real time information director **502** enforces security constraints. Any access to the data that bypasses the real time information director **502** might compromise data security. Real time information director **502** is suitably flexible to manage access to the data.

[0150] Referring to **FIGURE 5A**, the data services handler **500** performs an identity management function, including registering a user or customer. The data services handler **500** is formed as a request-response model that can be enriched with a variety of services that are integrated with the real time information director **502**. The document metadata **504** implements services that can add more information to the resulting document. One common use of these features is to provide a recommendation, such verification of identity.

[0151] **FIGURE 5B** depicts the data services handler **500** in a multiple-level security application. Here, document metadata **504** includes policy metadata **506** that specifies agents **508** that are allowed access to data, and defines rules setting forth agent relationships **510** determining which agents have access to particular data along with requisites for attaining the data access.

[0152] The policy metadata **506** can be configured to enable operation of a security manager that identifies an agent on behalf of whom information is queried or updated, and identifies a consumer for whom information is requested. The agent is identified in a request for designated documents. An internal unique identifier is dynamically assigned to each consumer and agent as part of registration of the consumer and agent in the system. One or more external identifiers are mapped automatically to the internal identifier in each request.

[0153] The real time information director grants access to individual documents based on security policies that conform to the relationship between the agent and the affected consumer, and permissions granted by the consumer. Distinct security policies are specifiable for the individual documents. The security policies are customizable to recognize specific agent and consumer relationships expressed in a business model.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 209 of 302

[0154] Referring to **FIGURE 6**, a schematic block diagram illustrates an embodiment of a data services handler **600** that is configured with a data auditing capability. Many business environments may use an audit log, for example including an audit header **610** and audit details **612**, to record changes to the database **606**. In some cases the audit log only records summary information, such as the date and time that a document is received, along with key identifying fields of the document. In other cases, the audit log may record details of each modification, both before and after document access. Either level of auditing can be specified within document content metadata **604** as audit metadata **608**. The audit log may also include the identity of the user who submits a change, whenever the information is used to access the real time information director (RTID) **602**.

[0155] Referring to **FIGUREs 7A** and **7B**, schematic block diagrams illustrate a data services handler **700** capable of implementing a data cleansing functionality. **FIGURE 7A** depicts data cleansing on inbound documents. The real time information director (RTID) **702** includes Beans to cleanse and de-duplicate various types of master data, for example using a Loader Message-Driven Bean (MDB) **710**. The real time information director **702** includes a Supplier Manager **712**. Metadata **704** for a CREditor MASter (CREMAS) IDoc **714** directs the real time information director **702** to call the Supplier Manager **712** to cleanse and de-duplicate an inbound document. The CREMAS IDOC is a master data object for creditor information. Other applications may operate with other IDOCs such as DEBitor MASter (DEBMAS) and MATerial MASter (MATMAS) IDOCs. The Supplier Manager **712** invokes Trillium **716** via a Java Native Interface (JNI) to perform the cleansing and identify possible matches. A vendor or supplier **720** is inserted into the database **706** with the cleansed name and address by the real time information director **702**. The Supplier Manager **712** also adds the supplier **720** to a work queue if identified as a likely duplicate. Additional services within the Supplier Manager **712** enable the user to process the candidate duplicates, and confirm or deny the duplicates.

[0156] Let as assume that we want to distribute three types of master data objects, the material master, the creditor master and the debitor master. Let us also assume that we have four offices

[0157] **FIGURE 7B** shows the data services handler **700** in a configuration that cleanses and enriches inbound documents. Business rules **722** can be used to correct and enrich inbound documents prior to insertion into the database **706**. An aggregation engine **724** can be used to accumulate information to be inserted into the database **706**.

[0158] Data cleansing is one example of a value-added service that can be implemented using polymorphic metadata. An example is name and address cleansing, which is applicable to both customer master data and supplier master data. Document metadata **704** identifies name and address elements of the document. The elements are passed to the data cleansing service. The data cleansing service forms an abstract interface to third party data cleansing engines, such as Trillium engine **716**. The data cleansing service returns a name and address with normalized spelling, titles, street abbreviations, validated zip codes, etc. Supplier and customer data are inserted into the data store with cleansed names and addresses.

[0159] The data cleansing service also queues the supplier and customer information for detection of duplicates. Duplicates frequently arise in source data. Duplicate detection is used in RTSC because the same suppliers and customers appear in multiple SAP instances. The duplicate detection logic identifies likely duplicates. Logic is typically configurable to forward likely duplicates for human confirmation. Confirmed duplicates are cross-referenced in the database.

[0160] Subscriptions can use the cross-reference information in a variety of ways, such as limiting subscription to the authoritative instance of a master data document. The subscription filters out duplicate versions of the same supplier document emanating from multiple SAP instances. Subscription can also be constrained to subscribe to all documents, for example shipments, coming from a particular supplier identified in various SAP instances, or to collect data from various SAP instances related to a particular supplier into a single document. Subscription can further be used to enrich documents that reference a particular supplier, for example shipments, with information drawn from the master instance of the supplier.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 211 of 302

[0161] Referring to **FIGURE 8**, a schematic block diagram illustrates an embodiment of a data services handler **800** that implements business rules handling. A business rules engine **810** may be implemented in various configurations. In a specific example, real time information director **802** can host a Blaze Advisor™ business rules engine **810**. The real time information director **802** enables access to data that drives business rules **820**. Typically a business object model is constructed to support Blaze. The business object model enables access to data, performs computations on the data that are relevant to the rules, and supports execution of decisions made by application of the rules. The real time information director **802** includes a pre-built business object model that is operational without customer modification. The business object model gives direct access to the data described by the document metadata **804** and also enables access to aggregates defined for a specified document type using Aggregate Wizards. The business object model may invoke the scoring engine using any scoring model relevant to the document.

[0162] With a capability to create custom aggregate functions, any calculation likely to be used by business rules can be performed without modifying the business object model, a capability that can tremendously shorten the time expended in building a solution using business rules **820**.

[0163] Business rules **820** can be utilized in both a request-response model, and in an event-driven model.

[0164] Referring to **FIGURE 9**, a schematic block diagram illustrates an embodiment of a data services handler **900** that operates according to a request-response model. Client applications **920** may also request a document upon demand, using a request-response model. In the request-response model, the user invokes a servlet **922** for assistance, identifying information that can be used to assemble the document used by the business rules. For example, the information might include the name of a customer, and the identifier of a desired product that is currently back-ordered.

[0165] In the illustrative embodiment, the servlet **922** passes the requested document type, along with any identifying parameters, to a session bean **924**. The real time information director (RTID) **902** assembles the requested document, using the document metadata **904** in a manner of operation similar to triggering by a subscription. The

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 212 of 302

servlets **922** may be advertised as web services, including the supported document definitions.

[0166] Referring again to **FIGURE 8**, a servlet **830** calls the real time information director (RTID) **802**, requesting the product-cross-sell document, and passing the customer and product identification. The document metadata **804** directs the real time information director **802** to assemble past data about the customer's buying patterns and preferences, as well as information about the supply of comparable products, and pass the document to the business rules **820**. The business rules **820** insert into the document recommendations for products to cross-sell. The business rules **820** may use mining models that predict likelihood that the customer will buy certain products. Business rules may also consider supply and/or apply rules that describe current promotions. The completed document is returned to the servlet **830**, which displays recommendation on a web page. In the event-driven model, a subscription triggers the creation of a document that is rules-enabled. After document metadata **804** is used to assemble the data from the database **806**, business rules **820** can again insert recommendations into the document and direct the document to a particular queue for further action. For example, suspiciously large orders can trigger a subscription. The business rules **820**, possibly with assistance from scoring models **822**, and can determine likelihood that the order is mistyped, manipulated or fraudulent. The messages in a queue can be automatically processed by another application, or displayed to an authorized manager. A business rules capability thus enables intelligent, filtered, subscriptions.

[0167] Referring to **FIGUREs 10A** through **10K**, multiple schematic block diagrams illustrate various real time information director interfaces. **FIGURE 10A** shows the real time information director (RTID) **1000** as a "black box" that has a few simple programmatic interfaces **1002**. Input and output interfaces are in XML **1004**. Documents are defined in metadata **1006** and document content reflects a real-time data model **1008**.

[0168] **FIGURE 10B** shows programmatic interfaces **1002** including a servlet and stateless session bean **1010** that handles inbound and outbound documents synchronously. A message driven bean handles inbound documents asynchronously and a subscription service delivers outbound documents asynchronously via JMS, components that are deployed under WLS.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 213 of 302

[0169] **FIGURE 10C** depicts generic data model customizations. A user modifies data model (DDL) **1008**. The real time information director (RTID) **1000** reads SQL schema dynamically. The user creates content metadata to identify records that make up a document. The real time information director **1000** exports an XML schema **1012** for native documents **1014**. The user defines events to trigger custom subscriptions **1016**.

[0170] **FIGURE 10D** shows enrichments. Various enrichments can be invoked via document metadata. Enrichments can be invoked on inbound or outbound documents. The user extends the data model **1018** to capture inbound enrichments. Some enrichment systems, for example Customer Manager, have a dedicated data model. Some enrichment systems, for example Blaze, have dedicated metadata, for example business rules **1020**.

[0171] **FIGURE 10E** illustrates interface features that are specific to Real-Time Supply Chain (RTSC). The data model is the RTSC data model **1022** and the interface supports IDocs **1024** in addition to native documents. The metadata **1026** supports variant mappings for the same IDocs from different source systems. RTSC IDocs are based on a standard set of starting IDocs.

[0172] **FIGURE 10F** shows an HyperText Transport Protocol (HTTP) interface that can form a basis for the RTSC data model for receiving inbound IDocs. The interface supports variant mappings **1028** for same IDocs received from different source systems. VISTA delivery uses a native document. The servlet & bean interface **1030** that is used is obsolete, supporting updates only.

[0173] **FIGURE 10G** illustrates an interface that supports Electronic Health Record (EHR)-specific features. EHR can be implemented according to Veterans Health Administration (VHA) electronic medical records in a Computerized Patient Record System (CPRS) using a technical infrastructure that displays various clinical functions in a graphical user interface (GUI) format. The business data model **1032** is specific to EHR. Other features are applicable to other industries.

[0174] **FIGURE 10H** shows a request-response interface to the real time information director that handles both updates and queries.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 214 of 302

[0175] **FIGURE 10I** illustrates an interface embodiment that is configured for auditing functionality. Metadata **1036** contains audit header and detail tables. Metadata **1036** also indicates audit options per document and audit queries **1038**, as well as document header contents in conformance with an auditing data model **1040**.

[0176] **FIGURE 10J** depicts an embodiment of a security interface based on a security data model **1042** and using document security options and security policies implemented in security metadata **1044** including security tables that record agent/consumer relationships, agent groups, roles that are granted rights based on agent/consumer relationships, and opt in/out exceptions. Metadata **1044** also indicates security constraints by documents, and security policy sets that are invoked.

[0177] **FIGURE 10K** shows a management interface **1046**. The real time information director **1000** is managed using standard J2EE and WLS mechanisms. Events are reported through WLS log. A management bean **1046** exposes performance counters and makes real-time configuration changes, such as deploying new metadata.

[0178] Referring to **FIGURE 11**, a schematic block diagram illustrates an embodiment of an information handling system **1100** that can be used for Zero Latency Enterprise (ZLE) applications and data handling system. The ZLE framework is supported by Hewlett Packard Company of Palo Alto, CA. The information handling system **1100** includes a real time information director **1102**, for example as described with respect to **FIGUREs 3A** and **3B**. The real time information director **1102** is a functional computation engine configured to enable real-time aggregation and real-time scoring. In an illustrative embodiment, the real time information director is a black-box application framework for deploying custom systems that perform real-time scoring and implementation of business rules. Typical uses of the real time information director are to make personalized offers and to advise of attempts to commit fraud.

[0179] The real time information director **1102** comprises an aggregation engine capable of computing aggregates in real-time from in-memory data that is dynamically cached during a session with an entity.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 215 of 302

[0180] The information handling system **1100** manages interactions using the real-time information director **1102**, a component that efficiently caches detailed entity data during interactive sessions, forwards the information to the real time information director to personalize the session, and make offers. The entity data is supplied during interactive sessions via one or more entity interfaces **1116**. In a common application, the entities can be customers who supply information relating to commercial transactions and inquiries. In various applications, the entities can be any sort of entity including persons, corporations, organizations, government agencies, and the like. In an illustrative embodiment, the real time information director **1102** is supplied by Hewlett Packard Company of Palo Alto, California. Hewlett Packard Company also has a ZLE Development Kit, familiarly known as the ZDK, used by HP professional services and partners to build custom ZLE applications that run on a NonStop™ Server platform **1110**. The NonStop Server platform serves as an information base that stores, for example, deployed models **1108**, customer data **1112**, and interaction data **1114**.

[0181] Among the application templates in the ZDK, two are highly useful in the real-time application of data mining. The real time information director **1102** hosts business rules **1106** that are implemented in a rules engine and makes the offers. The business rules **1106** may utilize scores obtained from mining models **1108**. A specific example uses a Blaze Advisor rules engine, which can supply offers based on business rules that are written and tested in Blaze Advisor Builder. Blaze Advisor can be licensed from Fair Isaac Corporation of San Rafael, California. Blaze Advisor is incorporated into the real time information director by adding appropriate Java™ Archive (JAR) format files obtained from Fair Isaac to the real time information director class path.

[0182] The real time information director **1102** returns offers based upon information supplied by interaction management. The illustrative real time information director **1102** has multiple component engines that derive offers from the input data. The engines work in various combinations and use business rules **1106** and/or data mining models **1108** to derive the offers.

[0183] The scoring engine scores information using data mining models built using a data mining engine **1118** and aggregates **1124** in the data mine **1120** creating models **1122** in the data mine **1120**. One suitable data mining engine **1118** is SAS Enterprise Miner™. Models generated by SAS Enterprise Miner are deployed to the NonStop SQL database using Genus Deployment Tool **1126** from Genus Software, Inc. of Cupertino, CA. The scores can be used within the Blaze Advisor business rules, as one criterion for making an offer. Usage of data mining models within business rules is not mandatory. Data is prepared on the NonStop Platform **1110** using data preparation tools **1128** and moved from the NonStop Platform **1110** to the data mine **1120** using the transfer tools **1128**. Data can be deployed from the data mine **1120** to the NonStop Platform **1110** using a model deployment tool **1126**.

[0184] Referring to **FIGUREs 12A** through **12E**, Unified Modeling Language (UML) diagrams are shown that describe operations and functionality of the real time information director. Document definitions determine behaviors of the real time information director. The document definitions are implemented in a concrete subclass of DocumentDefinition. A person who performs a customization operation writes the concrete DocumentDefinition subclass. A concrete DocumentDefinition subclass is instantiated only once.

[0185] **FIGURE 12A** is a content management diagram **1200**. Content metadata **1202** for a DocumentDefinition **1204** is defined by a DocumentRecord **1206**, which contains various RelatedRecords **1208**. RelatedRecords **1208** may contain other RelatedRecords **1208**, the only classes shown in the content management diagram **1200** that are visible to the persons performing the customization. A DocumentRecord **1206** or RelatedRecord **1208** typically refers to a table by name, causing a record definition (RecordDef) **1210** for the table to be obtained by reading a SQL schema **1212** of the table if the SQL schema has not already been read. RecordDefs **1210** contain a field definition (FieldDef) **1214** for each column of the table. Enrichment metadata specified in the DocumentRecord **1206** or RelatedRecords **1208** may trigger the creation of an enriched RecordDef **1210** containing additional FieldDefs **1214**, for example from a Joined record, or fewer FieldDefs **1214.**

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 217 of 302

[0186] The DocumentDefinition **1204** uses the metadata **1202** to construct the tree of RecordSets **1216**. The RecordSets **1216** integrate the content metadata, for example DocumentRecords, RelatedRecords, and enrichments nested beneath, with format metadata that is not shown in the diagram. Multiple aliased or variant document definitions, with different format metadata, can share the same content metadata **1202**. A separate tree of record sets is constructed from the same content metadata for each aliased or variant document definition.

[0187] A Dispatcher **1218** allocates and manages DocumentHandlers **1220** to process inbound documents and subscriptions. Each DocumentHandler **1220** is allocated a SQL connection to the database **1222**. The DocumentHandler **1220** includes a tree of RecordHandlers **1224**. Each RecordHandler **1224** is constructed and prepares applicable Java DataBase Connectivity (JDBC) statements based on information in the RecordSet **1216** and associated RecordDef **1210**. RecordHandlers **1224** either act on the appropriate Records **1226** in the inbound document to update the database, or construct the appropriate Records **1226** for the outbound Document **1228** after reading the database.

[0188] The diagram illustrates parallelism between data **1222**, document handlers **1220**, and metadata **1202**. One tree of RelatedRecords **1208**, with DocumentRecord **1206** as the trunk, is constructed for each *base* document definition **1204**. One tree of RecordSets **1216** is constructed per base, variant or aliased document definition. One tree of RecordHandlers **1224**, containing prepared statements, is constructed per database connection and DocumentHandler **1220**. One tree of Records **1226** is constructed per Document **1228**.

[0189] Referring to **FIGURE 12B**, a UML diagram illustrates an embodiment of an Insert function **1230**. Several of the classes shown in the content management diagram **1200** are abstract. The Insert diagram **1230** shows concrete classes involved in a Document Definition that inserts a document into the real-time data store.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 218 of 302

[0190] The Dispatcher **1218** parses and disassembles inbound eXtended Markup Language (XML) into the RecordsDocument **1228**. A name is identified during the process. In most cases, the name corresponds to the top- level tag of the document. However, in the case of Variant documents, the name of the document is identified using other criteria, such as the source system.

[0191] The Dispatcher **1218** looks up the DocumentDefinition **1204** corresponding to that name, and also looks for an available existing DocumentHandler **1220** for the underlying base document definition. If none currently exist, or all existing DocumentHandlers **1220** for that base document definition are currently in use, the Dispatcher **1218** creates a new DocumentHandler **1220**. Attributes of the concrete document definition are read by the Dispatcher **1218** to determine the concrete type of DocumentHandler **1220** to construct that can handle any inbound document.

[0192] In the illustrative example, a new patient is registered. A PatientRegistration **1232** document definition is self-identified as an insert document. The Dispatcher **1218** accordingly creates an InsertDocumentHandler **1234** for the PatientRegistration **1232**. The InsertDocumentHandler **1234** creates a RecordInserter **1236** for the table named in the DocumentRecord **1206**. That RecordInserter **1236** creates RecordInserters for each RelatedRecords **1208** that is child to the DocumentRecord **1206**. Each RecordInserter **1236** in turn creates RecordInserters for any RelatedRecords **1208** that is child to the corresponding RelatedRecords **1208**. Each RecordInserter **1236** holds the prepared statement to insert records into a particular table. The complete prepared statement is determined by the RecordDef **1210** for that RecordSets **1216**.

[0193] The RecordDef **1210** for the top record - the patient record - is enriched using a UniqueId enrichment, indicating that the patient should be assigned a new internally unique identifier.

[0194] Referring to **FIGURE 12C**, a UML diagram depicts an embodiment of a Delsert diagram **1238**. The diagram **1238** shows concrete classes used to "Delsert" a document. Delsert is a contraction of "delete and insert" and is used when a document is a complete replacement for a previous instance of a particular document.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 219 of 302

[0195] The example illustrates processing of a sales order document. The Dispatcher **1218** obtains a DelsertDocumentHandler **1240** for an Order document. The DelsertDocumentHandler **1240** first deletes rows associated with any previous version of the Order **1242**. If the order is not previously in existence, the DelsertDocumentHandler **1240** simply inserts the new order. The DelsertDocumentHandler **1240** does not replace the previous document, however, if the order is more recent than the document currently received. Due to latencies, documents can be received out of order.

[0196] The DelsertDocumentHandler **1240** creates a RecordDelserter **1244**, which creates child RecordDelserters, grandchildren, and the like. The RecordDelserter **1244** contains a prepared delete statement in addition to the prepared insert statement inherited from the RecordInserter **1236**.

[0197] The order **1242** also updates a history table that tracks schedule changes in the line items of the order. The HistoryRecord metadata class, a subclass of RelatedRecords **1208**, instructs the parent RecordDelserter **1244** to create a RecordInserter **1236** instead for that one table.

[0198] Referring to **FIGURE 12D**, a UML diagram illustrates an embodiment of a query diagram **1246**. The query diagram **1246** shows concrete classes used to query a document. Although not shown in **FIGUREs 12A** through **12C**, the real time information director automatically creates a QueryDocumentDefinition **1248** for each update document that is defined by a customizer. The QueryDocumentDefinition **1248** defines input criteria for the query, and has a reference to the corresponding update document definition that defines the content to be returned. The query diagram **1246** shows how the same user-specified metadata **1202** is used to define both inbound and outbound data; both updates and queries.

[0199] A document is returned as a result of each update that satisfies the XML schema **1212** of the corresponding query document definition. The query document definition contains keys of the updated document. If the same document is submitted to the real time information director as input, a query operation is triggered, returning all data related to the topic. The document has a top level tag that is the same as the

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 220 of 302

corresponding update document, except that "_R" is appended to the name. The "_R" identifies a response to an update or a request for a query.

[0200] In the example, the inbound document is an order query. The top tag in the document is ORDER_R. The Dispatcher **1218** obtains a QueryDocumentHandler **1250** corresponding to this document. However, the QueryDocumentHandler **1250** is not created from the QueryDocumentDefinition **1248**, but from the Order document definition **1242**. The QueryDocumentHandler **1250** creates a RecordAssembler **1252** which creates child RecordAssemblers, grandchildren, and the like. The RecordAssembler **1252** contains a prepared select statement with clauses determined by the enriched RecordDef **1210**.

[0201] The QueryDocumentDefinition **1248** contains a RequestedRecord **1254** in place of the DocumentRecord **1206**. RequestedRecord **1254** is a subclass of DocumentRecord **1206**. RequestedRecord **1254** creates a dedicated RecordDef **1210** that does not correspond to a physical record in the database **1222**. For every update document a QueryDocumentDefinition **1248** is automatically created. The automatically generated RequestedRecord **1254** has columns that correspond to the keys of the table referenced by the DocumentRecord **1206** of the update DocumentDefinition **1204**.

[0202] The customizer may not allow queries of selected documents. Two sets of options may be customizer-specified in the default constructor of a concrete DocumentDefinition **1204**. One applies to the update document. Another applies to the query document. "Invalid" can be included in either set of options to indicate that the update or the query is invalid.

[0203] The customizer may also explicitly create a QueryDocumentDefinition **1248** by writing a subclass of QueryDocumentDefinition **1248** with a default constructor. The constructor references an existing update DocumentDefinition **1204** to define contents to be returned by the query. The QueryDocumentDefinition **1248** must include a RequestedRecord **1254** in place of the DocumentRecord **1206.**

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 221 of 302

[0204] Referring to **FIGURE 12E**, a UML diagram shows an embodiment of an Upsert operation **1256** including concrete classes used to "Upsert" a document. Upsert is a contraction of "update or insert" and is an operation used when a document adds to existing information on a subject without necessarily representing all information on that subject.

[0205] In the illustrative case a personal data document is processed and contains information including name, addresses, telephone numbers, dependents, and the like for some customer or patient. The Dispatcher **1218** obtains an UpsertDocumentHandler **1258** for a PersonalData document **1260**. The UpsertDocumentHandler **1258** first fetches all existing personal data for the person from the database **1222**. The fetched data is compared with the data in the inbound document. Any rows found to be new to the inbound document are inserted into the database **1222**. Any rows found to be changed are updated to the database **1222**. No further operation against the database **1222** is taken for rows that are unchanged or are omitted from the inbound document.

[0206] The UpsertDocumentHandler **1258** creates a RecordUpserter **1262**, which creates child Upserters, grandchildren and the like. The RecordUpserter **1262** contains a prepared update statement, in addition to the prepared insert statement inherited from the RecordInserter **1236**. The UpsertDocumentHandler **1258** also creates a RecordAssembler **1252** that enables reading of the current data for the document.

[0207] Some enrichments analyze both the old and new data when upserting a document. For example, aggregate enrichment causes totals, counts and other aggregates to be rolled up to a parent record. Aggregate enrichment includes both new and existing rows. The instance-ID enrichment monitors instance-IDs in use by old child records before assigning instance-IDs to new sibling records.

[0208] Referring to **FIGURE 13**, a Unified Modeling Language (UML) diagram is shown that illustrates various enrichment classes. Both the DocumentRecord **1206** and the RelatedRecords **1208** classes, along with associated subclasses, contain an optional Enrichment **1300**. Enrichment is an abstract class. Enrichments can be combined in two ways. First, all standard enrichment classes are subclasses of the abstract Enrichment subclass, ChainableEnrichment **1302**, enrichments that can be chained by nesting each

within an enclosing enrichment. Second, the customizer may construct a CompoundEnrichment **1304**, and pass an array of Enrichments to CompoundEnrichment **1304**.

[0209] Enrichment defines multiple template methods that are called at various points during program execution. The template methods are passed various pieces of information. The concrete enrichment classes can implement any of the template methods.

[0210] An *enrichInbound* method is passed a Group object after the real time information director has parsed the XML document. The Group object contains a hashtable of string values that can be obtained by the name of the column. A column that is not present in the input, is not present in the hashtable. The Group also contains a vector of Groups representing the children of the Group.

[0211] OrderLineEnrichment **1306** is an example of a custom enrichment class that implements *enrichInbound* and is passed the group representing an order line. OrderLineEnrichment **1306** reconciles information between the order line and various schedule line details found in its child groups.

[0212] SystemKey **1308** is a standard enrichment that implements *enrichInbound* and is passed a KeyDispenser **1310**. KeyDispenser **1310** dispenses unique 64-bit numbers for keys to a record. If several document definitions use the same set of system-generated keys, KeyDispenser is constructed once in your metadata classes and has all the document definitions referring to the same KeyDispenser **1310**. The KeyDispenser **1310** uses the KeyManager **1312**, a singleton class. The KeyManager **1312** allocates a block of keys for the KeyDispenser **1310** and reads and updates the system keys table to ensure that no two instances of the real time information director allocate the same key values. The block of keys is sized sufficiently large so that calls to the KeyManager **1312** are infrequent, otherwise multiple threads of DocumentHandlers and real time information director instances are likely to block while waiting on the KeyManager **1312**.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 223 of 302

[0213] The *enrichRecordDef* method is passed a RecordDef **1210** and returns a RecordDef **1210**. The method enables the enrichment to replace the RecordDef **1210** with an enriched RecordDef **1210**. The enrichment is called when the metadata is constructed. LeftJoin **1314**, ColumnList **1316**, OrderBy **1318**, and ForeignKey **1320** are examples of standard enrichments that implement *enrichRecordDef*. When the enrichments are compound or chained, the RecordDef **1210** can be replaced multiple times.

[0214] Some Enrichments **1300** have a defined context or state. For example, an enrichment may use a prepared SQL statement. The Enrichments **1300** are instantiated as part of a DocumentDefinition **1204**, within the corresponding RecordSets **1216**. The Enrichments **1300** cannot hold state because multiple threads, DocumentHandlers, can concurrently use the same DocumentDefinition **1204**. Therefore, each RecordHandler **1224** has an array of EnrichmentContexts **1322**. One of the template methods that an enrichment **1300** may implement is *instantiateEnrichmentContext*. The method is called at the time that a RecordHandler **1224** is created for a new DocumentHandler **1220**. The RecordHandler **1224** is passed to the *instantiateEnrichmentContext* method. An enrichment that uses state constructs the particular EnrichmentContext **1322** subclass, and calls the *addEnrichmentContext* method of the RecordHandler **1224** to store the state.

[0215] The diagram shows two contextual enrichments including ForeignKey and Aggregates, each of which has a corresponding enrichment context class.

[0216] The EnrichmentContext **1322** has template methods *enrich* and *enrich2*. Each is passed a Record. Records are generated from the Groups after parsing of the inbound document is complete. Unlike a Group, which only contains named Strings, a Record includes an array of Fields, each with specific data type information, obtained from the database or from enrichments that enrich the RecordDef **1210**. A field can obtain a value from the corresponding Group, or from any ancestor Group that has a named value corresponding to the name of the field.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 224 of 302

[0217] In the UpsertDocumentHandler **1258**, *enrich* is called before the previous data is read from the database. *Enrich2* is called after the previous data is merged from the previous document. ForeignKey **1320** implements *enrich* to resolve the enrichment before reading old data. Aggregates and InstanceId implements *enrich2* because the aggregates and instance IDs take into consideration *all* the data, both new and old.

[0218] ForeignKeyEnrichmentContext **1324** contains the prepared statement used to lookup the internal ID of a referenced record using the external IDs passed in the inbound document. ForeignKeyEnrichmentContext **1324** also contains the prepared statement used to lookup the external ID of a referenced record using the internal ID for an outbound document. The enrichment is often used when the referenced record has an artificial key, assigned using the SystemKey enrichment **1308** in some other document. The enrichment has other uses. For example, a patient event can be stored in the database with the system key of the patient. But the patient event document uses the external ID of the patient. The ForeignKey enrichment **1320** substitutes the system key for the external ID in the inbound document, and vice-versa in the outbound document, by reading the Person table that catalogues all the external ID / system key mappings.

[0219] The Aggregates enrichment **1344** contains an array of AggregateDefinitions **1326** that have names reflecting common aggregate functions, such as Sum **1328**, Min **1330**, Max **1332**, Mean **1334**, and Count **1336**. The Aggregates enrichment **1344** instantiates the AggregatesEnrichmentContext **1340** and uses the AggregateFactory **1342** in the RecordSets **1216** to construct an Aggregate object **1338** from each AggregateDefinition **1326**, to populate the AggregatesEnrichmentContext **1340**.

[0220] Referring to **FIGURE 14**, a UML diagram illustrates an embodiment of IDoc format metadata, depicting a technique for implementing format metadata for Systems, Applications, and Products (SAP) IDocs.

[0221] The XMLInputStream **1400** scans an input stream for eXtended Markup Language (XML) begin and end tags. Each begin tag causes a Tag object **1402** to be pushed onto a stack. Each end tag causes a Tag **1402** to be popped off the stack. Each Tag **1402** is associated with a TagDefinition **1404** at the time the tag **1402** is pushed onto the stack. The stack is initialized with a Tag that points to RootTag **1406**. RootTag **1406**

is a TagHash **1408** and contains all the top level tags that have been identified in all document definitions. Each DocumentDefinition **1204** causes an associated top tag to be added to RootTag **1406**.

[0222] When a tag is pushed onto the stack, a corresponding TagDefinition **1404** is searched by name from the TagHash **1408** that is associated with the last defined Tag pushed onto the stack. If no matching TagDefinition **1404** is found, then the new tag is considered undefined. Subsequent tags skip over the undefined tag when searching for the definition, thus treating undefined tags as non-existent tags.

[0223] TagDefinitions **1404** for complex elements, for example segments in IDoc terminology, derive from TagArray **1410**. TagArray **1410** contains an ordered list of TagDefinitions **1404**,in addition to the hash table inherited from TagHash **1408**. The list order determines the order in which tags are written to an XMLOutputStream **1412**.

[0224] Tags for simple elements derive from TagDefinition **1404** without inheriting from TagHash **1408** or TagArray **1410**. No tags can be nested under simple elements.

[0225] QualifiedSegment **1414** is Segment **1416** having a TagArray **1410** populated with SegmentQualifiers **1418**. SegmentQualifier **1418** contains a dedicated internal TagArray **1410**, which defines elements for the QualifiedSegment **1414** when the Qualifier applies. The QualifiedSegment **1414** searches associated SegmentQualifiers **1418** after parsing the first nested tag, uses, then replaces the associated definition on the stack with the TagArray **1410** associated with the SegmentQualifier **1418**, altering handling of subsequent tags within the segment.

[0226] The top tag in any IDoc is declared as an IDocSegment **1420**. The top tag associates the document with a DocumentDefinition **1204**. The IDocSegment **1420** always contains an IDoc tag **1422**, for example <IDOC BEGIN=1>. The IDocSegment **1420** constructs the IDoc tag **1422** automatically. Under IDoc Tag **1422** is HeaderSegment **1424**, for example <EDI_DC40>. After parsing HeaderSegment **1424**, the HeaderSegment **1424** passes to the DocumentDefinition **1204** to determine whether fields in the header match the criteria for any registered variants of the document. If so,

the IDocSegmentTag is replaced on the stack with the IDocSegmentTag for the variant document in a process similar to that performed by a QualifiedSegment **1414**.

[0227] RecordSegment **1426** and associated subclass IDocSegment **1420** are associated with a physical table. When the document definition is instantiated, RecordSets generated from the content metadata, for example DocumentRecord and RelatedRecords, are cross-linked to the RecordSegment tag. Each time a RecordSegment **1426** is parsed, a new Group is constructed, and made a child of the last Group triggered by a parent RecordSegment **1426**.

[0228] FieldElement **1428** causes a string value to be added to the hash table in the Group. The key under which the string is stored is the name of the table column identified in the metadata. The value of the string is parsed from the element. ParentElement **1430** is similar, except that the string is put into the hash table of the parent Group. FieldsElement **1432** causes the same value to be put into the Group under multiple names. Other format metadata classes perform similar variations. A FieldElement **1428** may have a Formatter object **1434**. Formatter **1434** is an abstract class. Concrete formatters modify the string that is put into the hash table in various ways.

[0229] TagDefinition **1404** has a number of template methods that are called by XMLInputStream **1400**, such as *begin* that is called when the begin tag is parsed, *data* that is called when the data of an element has been output, and *end* that is called when the end tag is parsed. The tag is passed on all of methods. The tag has a pointer to the next tag on the stack. All of the behaviors described above are implemented by overriding these template methods in the various subclasses of TagDefinition **1404**. Each of the TagDefinition **1404** classes manipulates the tag and/or the tag stack in various ways.

[0230] Segments may have a SegmentFormatter **1418**, another abstract class. Custom SegmentFormatters can modify the contents of the Group in unlimited ways. SegmentFormatters are called after the end tag of the segment has been parsed.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 227 of 302

[0231] TagDefinitions **1404** write tags to an XMLOutputStream **1412**. The TagDefinition **1404** has a template method called *assemble* that is passed the tag and the XMLOutputStream **1412**. Each TagDefinition **1404** contains information for writing a corresponding tag. TagArrays **1410** write each element or nested group in the Group, in the order specified by the TagArray **1410**. Formatters **1434** and SegmentFormatters **1436** are called during tag writing to undo the operations that are performed on the inbound document, if possible.

[0232] In an illustrative embodiment, real time information director (RTID) metadata can be mostly implemented in Java. As an object-oriented language, Java supports polymorphism. Descriptive metadata is usually expressed as parameters passed to Java constructors. A constructor in Java is invoked using a *new* statement. The constructor for each metadata class has parameters applicable to that object. For example,

```
new Circle(16);                    // diameter = 16
new Rectangle(10, 8);              // length = 10, width = 8.
```

[0233] New statements can be written or modified that define the metadata. A behavior is implemented within the other methods of each metadata class. Implementation of the behaviors is transparent to a user.

[0234] Systems can be implemented with metadata written in Java. In Java implementations, procedural code is not written. The main Java statement is *new* and the statements are relatively concise. Java metadata enables rapid evolution since metadata is represented in the language of implementation, facilitating enhancements and response to special requirements.

[0235] Alternatively, metadata can be represented in eXtended Markup Language (XML), which is also polymorphic. However, XML describes but does not implement behavior so that implementation or translation to another language, such as Java, would extend development time without avoiding complexity.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 228 of 302

[0236] The illustrative Real-Time Supply Chain (RTSC) uses two kinds of metadata including content metadata and format metadata. Baseline metadata definitions map to a baseline data model and may be extended to support specific implementations. Extensions may be implemented to create native documents for new or existing data models, create a new Interchange Document (IDoc) mapping that is not included in the baseline, and map a single IDoc type multiple times for integrating different SAP source systems. Additional purposes for implementing extensions include complying with XML Schema validation, inserting additional formatting and/or unformatting to an existing mapping, capturing additional information in the data store, possibly leading to modification of the data model. Extensions may be further implemented to adapt to changes in database schema or data model, for example, adaptations to changes in table column names, table primary keys. Similarly, extensions can be implemented to address new tables in the data model for a document.

[0237] Various suitable techniques may be used to extend metadata. In one example, the data model is optionally updated on the NonStop server to enable usage of new documents and changes to an existing data model. Changes can be made to metadata, for example using a Java Interface Development Environment (IDE) on a workstation. Metadata is compiled and tested on the workstation, for example using a stand-alone test program. The new metadata is deployed to the real time information director (RTID) server. Monitoring of a real time information director log on the NonStop server enables analysis and verification of metadata operation.

CREATING A NEW NATIVE DOCUMENT MAPPING

[0238] The real time information director can generate native documents as eXtended Markup Language (XML) representations of data in the NonStop SQL database. Typically the native documents are used by an application that consumes data from the NonStop SQL database, such as subscriber applications. Consumers of native documents can use the XML Schema for native documents to validate received documents.

[0239] Metadata for a new native document can be created by determining whether inbound documents are delserted or upserted, and determining information to be accessed from the database. Information to be accessed is determined by identifying the tables, and

the relationship between the tables, that contain the desired data. Metadata is further created by forming the data model if not currently in existence. For example, a native document can use the data model for an IDoc. Once the data model is formed, the XML Schema is created for the native document.

[0240] For inbound native documents, the real time information director (RTID) determines whether documents are delserted or upserted into the database. If the input documents always contain complete information, for example for different versions of the same document, the current document is a superset of the older versions of the document, then the document is appropriately delserted. In a delsert operation, the real time information director first deletes all existing records related to the current document, and then inserts the current document into the RTSC data store.

[0241] However, delsert does not delete records inserted in historical tables, including tables defined with HistoryRecord in the content metadata. The purpose of the historical table is to accumulate a historical trail for all transactions, implying that inbound data contains unique values for the primary key columns of the historical table. In the historical view relationship, a table always accumulates records, as opposed to event tables in which data can be updated or deleted. The same records that are updated in event tables are accumulated in historical tables, thereby forming an historical view of the data.

[0242] If the input documents are updates, for example for different versions of the same document so that the current document contains different data than the older versions of the document, then the document is appropriately to upsert. In the upsert operation, the real time information director updates any existing records related to the current document. If no such records exist, then the current document is inserted.

[0243] Once the mapping information exists, metadata is created by creating a new Java class in a suitable package, for example com.hp.rtsc.documents. The class is conventionally but optionally named after the document name. For example, for creation of metadata for a native INVOICE document, the name of the Java file can be "NativeInvoices.java". The new Java class extends the NativeDocumentDefinition class.

The call to the superclass constructor may include a string of options. No two documents can have the same top-level tag.

[0244] Once the new Java class is created, Java metadata objects are created for tables that have been analyzed, first by creating document content metadata objects (DocumentRecord, RelatedRecords, SplitRecords or HistoryRecord.), then creating Links objects to describe fields used to join two tables.

[0245] After creation of the new Java class, the Java source file is compiled, tested, and updated Java class files are deployed to the RTSC runtime environment according to instructions described herein for compiling metadata and deploying Metadata sections. To test inbound native documents, a dispatch and/or reselect program may be used. To test outbound native documents, a Select stand-alone program may be used. To test both inbound and outbound native documents, a Reselect program may be used.

[0246] An XML Schema can be created for the native document using an Export utility.

CREATING A NEW INTERCHANGE DOCUMENT (IDOC) MAPPING

[0247] To create metadata for a new IDoc that is not handled by the RTSC baseline, the information being captured is analyzed and the data model is updated to hold the additional information. As for a native document, whether inbound documents are delserted or upserted is determined. The mappings to be created to store the information in the data store is analyzed and determined, fundamentally to identify elements in the document to be mapped to particular columns in the database.

[0248] When an IDoc mapping is created for the first time, a Base Document is also created, defining for the first time the content and format metadata suitable and appropriate for the IDoc. The data model appropriate for the content metadata is also created.

[0249] Once the mapping information is available, multiple actions can be performed to create the metadata. In one embodiment, a new Java class is created, for example in a

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 231 of 302

package com.hp.rtsc.documents. Conventionally but optionally, the class is named after the IDoc name. For example, metadata created for an ORDERS05 IDoc can be named the Java file "Orders05.java". The new Java class extends the BaseDocumentDefinition class. The call to the superclass constructor may include DocumentDefinition.UPSERT. No two documents can have the same top-level tag except for variant IDocs.

[0250] In a second action of the illustrative embodiment, Java objects are created for the analyzed mappings. Document format metadata objects (IDocSegment, RecordSegment, FieldElement, and the like) are created along with content metadata objects (DocumentRecord, RelatedRecords, SplitRecords), and Links objects that describe fields used to join two tables.

[0251] In a third action of the illustrative technique, the Java source file is compiled, tested, updated Java class files are deployed to the RTSC runtime environment in accordance with instructions for compiling and deploying metadata described hereinafter. To test inbound IDocs, a dispatch or reselect program can be used. To test outbound IDocs, a Select stand-alone program can be used. To test both inbound and outbound IDocs, a Reselect program can be used in accordance with the description for testing metadata.

RESOLVING CONFLICTS BETWEEN SOURCE SYSTEMS

[0252] Mappings for an IDoc type typically already exist, for example ORDERS05 IDoc. Mappings include content and format metadata, and are based on an existing data model. A different mapping for the same IDoc type may be desired, for example in a case that inbound documents come from different SAP source system.

[0253] Different SAP source systems may have different mappings for the same IDoc. For example, two SAP instances can generate ORDERS05 documents with differences in data. In one instance, the documents may have a different top tag, ORDERS05 versus ORDERS05EXT. The real time information director presumes that the documents are two different IDocs since the top-level tag identifies the metadata for a particular IDoc. In another instance, the documents have the same top tag but different mappings. The real time information director presumes the two documents are the same

IDoc, since the top-level tag is the same. However, the documents should be handled differently because the mappings are not the same.

[0254] Regardless of the source system from which the real time information director receives the IDoc, the correct mappings are to be used to load the data in the database. The data stored in the database should look the same, regardless of the SAP source.

[0255] A new document definition is chosen as an alias or variant. The differences between different SAP source systems is analyzed and determined by obtaining sample IDocs and analyze differences. If the IDoc top tag is different than the top tag of the base document mapping, then a Variant Document Definition is created. If the IDOC top tag is the same as the top tag of the base document mapping, then an Alias Document Definition is created.

[0256] The Alias or Variant Document Definition mappings are created using the same actions as are used for creating a new IDoc mapping. However, some actions can be eliminated since a base document definition already exists that can serve as a starting point. Beginning with the base document definition, a new mapping is created. For example, the file Orders05.java can be copied and written to a file, for example named Orders05EXT.java. Assuming the top tag is different from the base definition top tag, then Orders05EXT.java can be edited and the class definition changed to extend AliasDocumentDefinition instead of BaseDocumentDefinition. The top-level tag can be changed to ORDERS05EXT in a form, as follows:

```
private static final IDocSegment idocSegment =
        new IDocSegment("ORDERS05EXT", "SALES_ORDER", new Segment[] {
        IDOC.HEADER,
        . . .
        }
```

[0257] The definition of the DocumentRecord content metadata is removed. Instead, the superclass constructor with the name of the base document which contains the content metadata on which this IDoc is based is called, for example "ORDERS05". Then

Orders05EXT.java source file is compiled, tested, and the updated Java class files are deployed to the RTSC runtime environment according to instructions herein for compiling metadata and deploying metadata sections.

[0258] To test inbound IDocs, Delsert or Disassemble stand-alone programs are used. To test outbound IDocs, the Select stand-alone program is used. To test both inbound and outbound IDocs, the Reselect program is used in accordance with the metadata testing instructions described herein. The mapping can be tested with both base document sample IDocs (ORDERS05) and with the new alias sample IDocs (ORDERS05EXT). Similar steps can apply to create a VariantDocumentDefinition subclass.

PASSING XML SCHEMA VALIDATION

[0259] Metadata may be insufficient, without modification, to enable validation of documents generated by the real time information director against the XML Schema for a particular IDoc. An XML Schema file can be used to verify that an XML file is valid. Content of Markup Language (ML) documents generated by the real time information director depends on the order in which the format metadata is defined.

[0260] Validation of an XML document against the XML Schema can be implemented using multiple actions. First, IDoc schemas are obtained from the SAP website. The schema is saved as an .XSD (XML Schema Definition) file. Second, a sample XML IDoc file is generated with the Real time information director (RTID) using the Select standalone program in accordance with the metadata testing description herein. Third, the XML IDoc file is opened using XML Spy. The file can be viewed in "Schema/WSDL Design view" mode. Fourth, the schema obtained from the first action is assigned to the XML document. For example, from XML Spy main menu, DTD/Schema -> Assign Schema... is selected to assign the schema. Fifth, the XML file is validated. For example, from XML Spy main menu, "XML -> Validate" is selected to validate the file. Sixth, error messages, if any, suggest metadata modifications. For example, current format metadata may generate the output file with the XML sequences in the wrong order so that the metadata is to be rearranged in the order defined by the XML Schema to pass the validation. Seventh, the format metadata is modified according to the error messages. Eighth, the Java source file is compiled, tested, and the updated

Java class files are deployed to the RTSC runtime environment in accordance with instructions for compiling metadata and deploying metadata sections described herein. To generate outbound IDocs in accordance with the second action, the Select stand-alone program can be used in accordance with the metadata test description herein. Ninth, the third to eighth actions are repeated until the XML IDoc passes the XML Schema validation.

ADDING FORMATTERS TO THE EXISTING MAPPING

[0261] Inbound IDoc data may contain values that are not stored in the database in the same format. One example is numeric values with leading zeros, and date or date-time fields that are formatted differently than SQL date or SQL date-time fields. For the real time information director to generate outbound documents from the database in the same format as the inbound documents, formatter metadata objects are specified.

[0262] An example of an element using the formatter object is a definition in Orders05.java, which uses a predefined formatter:

```
New FieldElement("POSEX", "SALES_ORDER_ITEM_NO", IDOC.ITEM_NO);
```

[0263] The predefined formatter ITEM_NO is defined in IDOC class as:

```
Formatter ITEM_NO = new LeadingZeros(6);
```

[0264] The class com.hp.rtsc.documents.IDOC defines the most commonly used formatters. Depending on the suitable formatting/un-formatting, the constructor of the Java object is modified to specify the formatter object. If the baseline does not include a suitable formatter object, a custom formatter can be created that subclasses the class com.hp.rtsc.metadata.formatters.Formatter.

[0265] Usually differences are apparent in comparison of inbound IDocs with outbound IDocs generated by Select or Reselect stand-alone programs. The Reselect program is useful for comparison tests because the program inserts a document into database and selects the same document from the database. Then, on comparison of the two documents, differences and the type of formatters to be added become apparent.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 235 of 302

Details of the comparison and response are disclosed in the metadata testing description hereinafter.

EXTENDING DATA STORE

[0266] Several actions are performed prior to updating metadata to capture additional information in the data store. Information to be captured is analyzed. Changes are made to the data model to hold the additional information. Mappings are analyzed, designed, and created to appropriately store the additional information in the data store.

[0267] Once the new mapping information is configured, the data store can be extended by opening the Java source file for the IDoc metadata mapping class. For example, to extend the ORDERS05 document, com\hp\rtsc\documents\Orders05.java file is edited. For each new element in the IDoc is to be captured in the data store, a statement is inserted to create the Java object that describes the mapping for the particular element. Inserted and added elements are check to ensure entry to the correct parent segment. For example, the following statement creates a FieldElement in the E1EDP01 segment, to map the "VPREI" element to the "NET_PRICE_AM" column in the database:

New FieldElement("VPREI", "NET_PRICE_AM").

[0268] The Orders05.java source file is compiled, testes, the updated Java class files are deployed to the RTSC runtime environment according to instructions for compiling metadata and deploying metadata sections herein.

[0269] To test the new mappings, first Disassemble is called, then Delsert stand-alone programs. Disassemble generates the output Records file containing data in format for insertion into the database. If the Records file is correct, Delsert is used to actually insert sample IDocs into the database, using the new mappings. The database is queried to ensure data is inserted correctly.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 236 of 302

CHANGING METADATA MAPPING

[0270] If any baseline mappings are not applicable to the implementation, metadata can be updated to suit the applications. Business characteristics are analyzed to determine suitable and appropriate mappings. The data model is modified according to the business characteristics. Once the mapping information is developed, multiple actions are performed to change the metadata. First, the Java source file for the IDoc metadata mapping class is opened. For example, if the metadata for the ORDERS05 document is updated, com\hp\rtsc\documents\Orders05.java file is updated. Second, for individual elements in the IDoc for which the mapping is to be changed, the Java object that relates to that element is located and changes made. For example, to map the element to a different column in the database table, the field name that is mapped to the IDoc element is updated. Similarly, to change a SegmentQualifier for a given element, SegmentQualifier is updated with the new qualifier. To add a new SegmentQualifier, the QualifiedSegment associated with the SegmentQualifier is located and a new SegmentQualifier is added. Third, the Orders05.java source file is compiled, tested, and the updated Java class files are deployed to the RTSC runtime environment in accordance with instructions for compiling metadata and deploying metadata sections described hereinafter. To test inbound IDocs, Delsert or Disassemble stand-alone programs are used. To test outbound IDocs, Select stand-alone program is used. To test both inbound and outbound IDocs, Reselect program is used in accordance with metadata testing described herein.

SQL SCHEMA UPDATING

[0271] For conditions and circumstances for which SQL Schema updating is desired, for example, column name changes, primary key changes, data type changes, and the like, metadata is also checked and updated. For example, if the name of a table column that is mapped to an IDoc element is changed, the Java object definition of the metadata element is also updated to specify the new column name. After editing of the Java source and updating of the metadata mappings are complete, the Java source file is compiled and the updated Java class files deployed to the RTSC runtime environment according to instructions for compiling and deploying metadata. Some SQL Schema changes to

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 237 of 302

existing tables, such as adding or dropping a column to the primary key, call for restarting the real time information director (RTID) server.

COMPILING METADATA

[0272] Metadata can be modified on a computing system of various types, for example a workstation, using a Java IDE or even Notepad. The Java IDE may be used to compile the metadata or a command-line compiler. In one embodiment, a process for compiling metadata includes actions of opening a DOS window, changing directory to the RTSC install directory, for example c:\rtsc, setting up an environment by running a c:\rtsc\bin\setenv.bat file, and compiling the metadata source file. For example, to compile the Orders05.java, the following command may be used to create the Orders05.class:

Javac com/hp/rtsc/documents/Orders05.java.

DEPLOYING METADATA

[0273] After metadata is compiled, multiple actions are performed to deploy the metadata to the NonStop server. Metadata is deployed prior to testing because test drivers use the DOCCLASS table from the NonStop SQL database to determine which metadata classes to load. In an illustrative embodiment, the class file is first transferred in binary mode using the File Transfer Protocol (FTP) to the com/hp/rtsc/documents folder in an RTSC OSS install directory on the NonStop server. Second, if the metadata definition is new, an entry in the DOCCLASS table is created. The RTSC framework looks up the table and determines the class to load to process the applicable IDoc. For example, to add Orders05 metadata, the following command SQL is used:

insert into =DOCCLASS values("com.hp.rtsc.documents.Orders05",
"Metadata for ORDERS05", current).

[0274] Third, the RTSC framework is restarted to pick up the updated metadata, or management tools are used to request the real time information director (RTID) server to reload the updated metadata.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 238 of 302

[0275] If the metadata is not deployed before testing, the new metadata class can be loaded by specifying the -c parameter to the test driver. All test drivers listed herein handle the -c parameter as in the following example:

Java Disassemble -c com.hp.rtsc.documents.NewMetadataClass <other params>.

[0276] A message is typically entered on a display screen, for example:

Load class com.hp.rtsc.documents. NewMetadataClass.

[0277] Finally, after deploying the new metadata on the real time information director (RTID) server, the server log is monitored to determine whether any errors are generated.

TESTING METADATA

[0278] RTSC framework uses two internal document formats including "Groups" and "Records". The internal document formats are a representation of IDoc XML elements that are stored in the database. The *Groups* internal document format represents the grouping or relationship between the various IDoc segments and elements.

[0279] The *Records* internal document format is similar to the database record in a format suitable for storage. The *Records* format facilitates verification that elements in the input XML are mapped to the correct columns in the correct table, a useful property for debugging SQL errors. For errors such as data truncations and numeric value out of range, the *Records* representation of the IDoc can be accessed and compared against the table schema definitions in the database. If a SQL error occurs and debugging of the error is sought, the *Disassemble* test driver can be run to create a *Records* representation of the IDoc XML. Then the *Records* file can be visually inspected against the database schema to debug the field that generated the SQL error.

[0280] Stand-alone programs listed hereinafter can and typically are run without using RTSC WebLogic services. The stand-alone programs directly access the NonStop SQL database to obtain a list of metadata classes from DOCCLASS table, and load the specified metadata classes from the local com\hp\rtsc\documents directory.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 239 of 302

EXAMPLE – METADATA NATIVE DOCUMENT DEFINITION

[0281] A native document is specified by creating a *new* class, giving the class a descriptive name. The class extends NativeDocumentDefinition. A DocumentRecord is constructed to specify the document content metadata. The constructor for the class calls the super class constructor with two parameters, a reference to the document record and the document name. An alternate constructor has three parameters including the document record reference, the document name, and string of options, for example including a security policy.

```
package com.hp.rtsc.documents;
import com.hp.rtsc.metadata.*;
Import com.hp.rtsc.metadata.internal.MetadataInconsistency;
import java.sql.SQLException;

/**
* Metadata class to define an ORDERS05 IDOC.
*/
Public class Orders05 extends BaseDocumentDefinition
. . .


    private static final IDocSegment idocSegment =
    new IDocSegment ("ORDERS05","SALES_ORDER", new TagDefinition[] {
    IDOC.HEADER
    New Segment("E1EDK01", new TagDefinition[] {
    . . .
    . . .
    Private static DocumentRecord documentRecord=
    new DocumentRecord("SALES_ORDER", new RelatedRecords[] {
    new RelatedRecords ("ROLE_SPEC_SLS_ORD_EVENT"),
    new RelatedRecords ("SALES_ORDER_DETAIL", new RelatedRecords[] {
    new SplitRecord ("SLS_ORD_SPEC_BUSINESS"),
```


SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 240 of 302

```
new SplitRecord ("SALES_ORDER_FLOW", new Links(new String[] {

"SALES_ORDER_NO","PRECEDING_DOCUMENT_NO",

"SALES_ORDER_ITEM_NO","PRECEDING_DOCUMENT_ITEM_NO"})),

 new RelatedRecords ("RL_SPEC_SO_DT_EV_OVRRD"),
 new RelatedRecords ("SLS_ORD_SCHEDULE_DT"),
 new RelatedRecords [] {
 new HistoryRecord("SLS_ORD_SCH_DT_HIST_STG")
 })
 })
 });

 // Metadata Constructor:
public Orders05 ()
throws MetadataInconsistency, SQLException
{
 super (documentRecord, idocSegment);
 }
```

[0282] DocumentRecord identifies the parent record of the document and is an object passed to the super class constructor. Parameters include the name of the table from which the parent record is drawn and an array of RelatedRecords. Keys from the table are used to select the record. Other records in the document are related to the record, directly using document keys and indirectly using other fields of the record or other related records.

[0283] Arrays of nested metadata are constructed as an array of RelatedRecords that is defined using the *new* statement, then constructing individual RelatedRecords. Nested metadata arrays are a direct representation of relationships between metadata. The RelatedRecords class selects one or more records from another table and includes parameters including the name of the table containing the related records and optionally a

links object defining a link of the table to a parent table or an array of related records nested under a current record. SplitRecord identifies a single record in another table that is related to the parent record and is a subclass of RelatedRecords.

[0284] HistoryRecord identifies a single record that is inserted into another table when a parent record is inserted or replaced. HistoryRecord is a subclass of RelatedRecords. HistoryRecord can be used to capture schedule changes for line items in an IDoc.

[0285] The real time information director (RTID) metadata leverages the SQL schema data definition language (DDL). The real time information director queries the database to determine information, for example name, datatype, and defaults, about the individual tables used by a document.

[0286] The eXtended Markup Language (XML) tags for a native document are generated automatically with the top tag assigned as the document name and following tags assigned as the names of tables or fields in the database. Data elements are delimited by tags with the field name. The real time information director includes a tool to export the XML Schema for the native document. The XML Schema can be imported into applications that support XML, and used to specify mappings within the applications. An example of XML excerpts for a native documents follows:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<NATIVEORDERS>
 <SALES_ORDER>
  <SALES_DOCUMENT_TY>YOR</SALES_DOCUMENT_TY>
  <PRECEDING_DOCUMENT_NO>0200114533</PRECEDING_DOCUMENT_NO>
  <CUSTOMER_PURCHASE_ORDER_DT>2002-10-18</CUSTOMER_PURCHASE_ORDER_DT>
  <INCOTERMS_2_TX>DELIVERED DUTY UNPAID</INCOTERMS_2_TX>
  <SOURCE_NM>ORDERS05SAPAPL</SOURCE_NM>
...
  <ROLE_SPEC_SLS_ORD_EVENT>
  <EVENT_SPEC_CUSTOMER_NO>ZHPF400</EVENT_SPEC_CUSTOMER_NO>
```

```
  <EVENT_SPEC_LANGUAGE_CD>E</EVENT_SPEC_LANGUAGE_CD>
...
  </ROLE_SPEC_SLS_ORD_EVENT>
  <ROLE_SPEC_SLS_ORD_EVENT>
  <EVENT_SPEC_TITLE_TX>Company</EVENT_SPEC_TITLE_TX>
  <EVENT_SPEC_LANGUAGE_CD>E</EVENT_SPEC_LANGUAGE_CD>
...
  </ROLE_SPEC_SLS_ORD_EVENT>
```

[0287] A unique IDoc is specified by creating a new class and descriptively naming the new class, for example to match the IDoc. The class is to extend BaseDocumentDefinition. The constructor for the class calls the super class constructor with two parameters including a reference to the document content metadata and a reference to the document format metadata. The document content metadata id defined in the manner of the native document definition. An example of a metadata excerpt ORDERS05 follows:

```
package com.hp.rtsc.documents;
import com.hp.rtsc.metadata.*;
import com.hp.rtsc.metadata.format.*;
import java.sql.SQLException;

/**
* Metadata class to define an ORDERS05 IDoc.
*/
public class Orders05 extends BaseDocumentDefinition {
      // XML IDoc Mapping Metadata.
      // This array of tags is declared separately,
      // since these tags occur both under <E1EDKA1> and <E1EDPA1>
      private static final FieldElement[] ORDER_EVENT_TAGS =
      new FieldElement[] {
            new FieldElement("PARVW", "PARTNER_FUNCTION_CD"),
            new FieldElement("PARTN", "EVENT_SPEC_CUSTOMER_NO"),
```

```
            new FieldElement("LIFNR", "EVENT_SPEC_SUPPLIER_NO"),
            new FieldElement("NAME1", "EV_SPEC_DOING_BUSINESS_AS_NM"),
    Private static DocumentRecord documentRecord =
    new DocumentRecord("SALES_ORDER", new RelatedRecords[] {
      new RelatedRecords("ROLE_SPEC_SLS_ORD_EVENT"),
      new RelatedRecords("SALES_ORDER_DETAIL", new RelatedRecords[] {
        New SplitRecord("SLS_ORD_SPEC_BUSINESS"),
        New SplitRecord("SALES_ORDER_FLOW", new Links(new String[] {
          "SALES_ORDER_NO","PRECEDING_DOCUMENT_NO",
          "SALES_ORDER_ITEM_NO","PRECEDING_DOCUMENT_ITEM_NO"})),
        New RelatedRecords("RL_SPEC_SO_DT_EV_OVRRD"),
        New RelatedRecords("SLS_ORD_SCHEDULE_DT",new RelatedRecords[]{
          new HistoryRecord("SLS_ORD_SCH_DT_HIST_STG")
        })
      })
    });

    // Metadata Constructor:
    public Orders05()
          throws MetadataInconsistency, SQLException {
          super(documentRecord, idocSegment);
    }
}
```

[0288] IDocSegment identifies the top level XML tag of an IDoc and normally has the same name as the IDoc. IDocSegment is a tag that is passed to the super class constructor. All other format metadata is nested under the IDocSegment object. Parameters of IDocSegment include the name of the XML tag, the name of the table to which the top record belong, and an array of segment tags nested under the tag.

[0289] IDOC.HEADER is the first tag under the new IDocSegment and represents an IDoc header.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 244 of 302

[0290] The IDoc class contains statically-initialized metadata objects that can be re-used in various IDocs. IDOC.HEADER is one object that is used in IDoc definitions. Other re-usable metadata objects can be added to the IDOC class or other classes. An example of an XML IDoc header is shown as follows:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<ORDERS05>
    <IDOC BEGIN="1">
        <EDI_DC40 SEGMENT="1">
            <TABNAM>EDI_DC40</TABNAM>
            <MANDT>016</MANDT>
            <DOCNUM>0000000001651543</DOCNUM>
            <DOCREL>46C</DOCREL>
            <STATUS>30</STATUS>
            <DIRECT>1</DIRECT>
            <OUTMOD>2</OUTMOD>
            <IDOCTYP>ORDERS05</IDOCTYP>
            <CIMTYP>ORDERS05</CIMTYP>
            <MESTYP>ORDRSP</MESTYP>
            <SNDPOR>SAPAPL</SNDPOR>
...
            <CREDAT>20030916</CREDAT>
            <CRETIM>040122</CRETIM>
            <SERIAL>20030916040120</SERIAL>
        </EDI_DC40>
```

[0291] The following XML IDoc segment excerpt illustrates three data segments:

```
<E1EDK01 SEGMENT="1">
    <ACTION>000</ACTION>
    <CURCY>JPY</CURCY>
    <WKURS>1.00000</WKURS>
    <ZTERM>J328</ZTERM>
    <BELNR>0300060645</BELNR>
    <VSART>01</VSART>
    <VSART_BEZ>STANDARD</VSART_BEZ>
    <RECIPNT_NO>W0J008399</RECIPNT_NO>
    <AUGRU>00</AUGRU>
    <AUGRU_BEZ>New orders</AUGRU_BEZ>
</E1EDK01>
<E1EDK14 SEGMENT="1">
    <QUALF>006</QUALF>
    <ORGID>ZD</ORGID>
</E1EDK14>
<E1EDK14 SEGMENT="1">
    <QUALF>007</QUALF>
    <ORGID>ZZ</ORGID>
</E1EDK14>
```

[0292] The next tag definition following IDOC.HEADER is a Segment tag. In an illustrative example, for ORDERS05 the initial segment is E1EDK01 and contains information for the top level record. Segment is SAP terminology for a record or group of related fields in an IDoc. The term segment is also used to refer to a complex XML element, for example an XML tag that does not directly introduce data but only other XML elements. As a result, the segment contains an array of nested tag definitions. Following the segment is a FieldElement that maps an element to a table column. FieldElement parameters include a tagName string, a columnName string, and an optional formatter. A FieldsElement maps the same XML element to multiple columns in a

database, for example in cases that the same data is stored by a different name in different tables. FieldsElement parameters include a tagName string, a columnNames string array, a fieldElements array, and an optional formatter. The following metadata excerpt illustrates an example of the start of an IDocSegment:

```
private static final IDocSegment idocSegment =
new IDocSegment("ORDERS05", "SALES_ORDER", new TagDefinition[] {
   IDOC.HEADER,
   new Segment("E1EDK01", new TagDefinition[] {
      new FieldElement("ACTION", "SALES_ORDER_ACTION_CD"),
      new FieldElement("CURCY", "CURRENCY_CD"),
      new FieldElement("WKURS", "PRICING_STATISTICS_EXCHANGE_RT"),
      new FieldElement("ZTERM", "PAYMENT_TERMS_CD"),
      new FieldElement("BSART", "CUSTOMER_PURCHASE_ORDER_TY"),
      new FieldsElement("BELNR", new String[]
      {"SALES_ORDER_NO", "PRECEDING_DOCUMENT_NO"}),
      new FieldElement("BSTZD", "PURCHASE_ORDER_NUMBER_DS"),
      new FieldElement("VSART", "SHIPPING_CONDITION_CD"),
      new FieldElement("VSART_BEZ", "SHIPPING_CONDITION_DS"),
      new FieldElement("RECIPNT_NO", "CUSTOMER_SOLD_TO_NO"),
      new FieldElement("KZAZU", "ORDER_COMBINATION_IN"),
   . . .
      new FieldElement("DELCO", "AGREED_DELIVERY_TM")
   }),
```

[0293] QualifiedSegment can be used to compress data, for example by mapping different occurrences of the same XML element to a number of distinct columns in the record of the database. QualifiedSegment is used on recurring segments for which a first element serves as a qualifier. QualifiedSegments are often used to represent different uses of similar data objects, such as dates, organization identifiers, and document numbers. SegmentQualifier maps one instance of QualifiedSegment based on one value of the qualifier element. Parameters for the SegmentQualifier may include qualifierValue, mappedTagName, columnName, or alternative qualifiers such as

FieldElement field and impliedField which can be used to insert a fixed value into a database column on input of qualifierValue. A segment qualifier can define multiple fields, for example the segment qualifier for E1EDK03 defines a time field as well as a date field. The following metadata excerpt illustrates an example of the start of a QualifiedSegment:

```
new QualifiedSegment("E1EDK14", "QUALF", new SegmentQualifier[] {
   new SegmentQualifier("001", "ORGID", "BUSINESS_AREA_CD"),
   new SegmentQualifier("006", "ORGID", "DIVISION_CD"),
   new SegmentQualifier("007", "ORGID", "DISTRIBUTION_CHANNEL_CD"),
   new SegmentQualifier("008", "ORGID", "SALES_ORGANIZATION_CD"),
   new SegmentQualifier("012", "ORGID", "SALES_DOCUMENT_TY"),
   new SegmentQualifier("010", "ORGID", "SALES_GP"),
   new SegmentQualifier("016", "ORGID", "SALES_OFFICE_CD")
}),
new QualifiedSegment("E1EDK03", "IDDAT", new SegmentQualifier[] {
   new SegmentQualifier("002", "DATUM", "REQUESTED_DELIVERY_DT"),
   new SegmentQualifier("005", "DATUM", "VALID_FROM_QUOTATION_DT"),
   new SegmentQualifier("006", "DATUM", "VALID_TO_QUOTATION_DT"),
   new SegmentQualifier("012", "DATUM","DOCUMENT_RECEIVED_SENT_DT"),
   new SegmentQualifier("022", "DATUM","CUSTOMER_PURCHASE_ORDER_DT"),
   new SegmentQualifier("025", new FieldElement[] {
      new FieldElement("DATUM", "RECORD_CREATE_DT"),
      new FieldElement("UZEIT", "ORDER_ENTRY_TM")
   }),
   new SegmentQualifier("023","DATUM","PRICE_AND_EXCHANGE_RATE_DT")
}),
```

[0294] RecordSegment maps a segment to a record in a table, according to parameters including tagName, tableName, tagDefinitions array, and optional ImpliedField. Many tag definitions can be nested within a RecordSegment. Segment defines a segment that does not introduce a new record into the database. Therefore, *segment* is to be used only when the segment does not occur more than once within any instance of the parent

segment. RecordSegment is a subclass of Segment. The following metadata excerpt illustrates an example of a RecordSegment:

```
new RecordSegment("E1EDP01","SALES_ORDER_DETAIL",new TagDefinition[]{
   new FieldsElement("POSEX", new String[]
   {"SALES_ORDER_ITEM_NO","PRECEDING_DOCUMENT_ITEM_NO"}, IDOC.ITEM_NO),
   new FieldElement("ACTION", "SALES_ORDER_ITEM_ACTION_CD"),
...
   new RecordSegment("E1EDP20","SLS_ORD_SCHEDULE_DT",new TagDefinition[]{
      new FieldElement("WMENG", "CONFIRMED_QT"),
      new FieldElement("EDATU", "SCHEDULE_LINE_DT"),
      new FieldElement("EZEIT", "DELIVERY_TM"),
      new Segment("ZEEDP20", new FieldElement[] {
         new FieldElement("ETENR", "SCHEDULE_LINE_NO", IDOC.LINE_NO),
         new FieldElement("LDDAT", "LOADING_DT"),
         new FieldElement("MBDAT", "MATERIAL_AVAILABILITY_DT"),
         new FieldElement("WADAT", "CURRENT_SCHEDULED_SHIP_DT"),
         new FieldElement("EDATU", "CUSTOMER_REQUESTED_RECEIPT_DT"),
         new FieldElement("WMENG", "REQUESTED_QT")
      })
   }),
   new RecordSegment("E1EDPA1","RL_SPEC_SO_DT_EV_OVRRD",ORDER_EVENT_TAGS),
```

[0295] Nested RecordSegments have automatic access to columns in parent and grandparent RecordSegments, a useful characteristic since key columns for parent records usually are included in child detail records. A column may be used in a table identified in the owning RecordSegment, tables identified in directly nested child RecordSegments, tables identified in indirectly nested descendant RecordSegments, and in any such combination. In a parent-child relationship, records in child tables link back to a record in a parent table.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 249 of 302

[0296] A RecordSegment may refer to more than one table. Tables can be horizontally segmented so that different columns of the same entity are stored in different tables. The same entities may be represented in different tables with different primary keys for mapping or lookup purposes. Each such segmented record is identified in the content metadata as a SplitRecord. Any column is stored into a segmented table having a column by the name of one or more of the segmented tables, or none of the segmented tables.

[0297] In a split-record relationship, data can be inserted as records into two different tables by splitting part of data into a first table and the remaining data in one or more additional tables.

[0298] In a lookup-table relationship, a portion of data that is inserted into a table is derived from data supplied by static lookup tables.

[0299] Mapped elements are put in a hash table keyed by column name with a separate hash table allocated for each parsed RelatedRecords and DocumentRecord. Before inserting in the database, the real time information director looks up each column in the owning RelatedRecords, then the parent, then the grandparent, all the way up to the DocumentRecord. Columns can be moved between different tables, or completely removed, without changing the mapping metadata. Accordingly, mapping is equally driven by the static mapping metadata and the dynamically-read schema of the database. ParentElement maps an element to a column in the parent table. QualifiedParentElement maps an element in a specific instance of a segment to a column in the parent table. FixedElement generates an element in subscribed IDocs that has an unchanging value. FixedElement may be used in IDOC.HEADER to populate fields of the outbound document header that are used to pass XML schemas published by SAP, but which are not captured in RTSC data store. Elements can be concatenated using PrependFieldElement and AppendFieldElement which map two elements into a single table column by concatenation. A distinct separator character is used to enable deconcatenation of the column when assembling the IDoc for subscriptions. SplitElement divides an element into two fields of the database.

[0300] Formatter classes can be used to convert element data from the IDoc document format, or custom format, to the database format or vice versa. For outbound documents, as elements are written out, formatter formats element data to the external document format. For inbound traffic, as elements are parsed the formatter unformats the element data to the database format. When possible, the formatter implements reformatting in both directions. A Formatter object can be passed as an optional parameter to the constructor of any FieldElement or subclass of FieldElement.

[0301] Automatic formatters can be applied to elements in IDocs based on the data type of the corresponding SQL field. A production order follows that includes many formatters:

```
new FieldElement("FREIZ", "REL_PERIOD_IN_DA", new LeadingZeros(3)),
new FieldElement("FTRMI", "ACTUAL_RELEASE_DT", IDOC.DASHED_DATE),
new FieldElement("FTRMS", "SCHEDULED_RELEASE_DT", IDOC.DASHED_DATE),
new FieldElement("GAMNG", "TOTAL_ORDER_QT"),
new FieldElement("GETRI", "CONFIRMED_ORDER_FINISH_DT", IDOC.DASHED_DATE),
new FieldElement("GEUZI", "CONFIRMED_ORDER_FINISH_TM", IDOC.TIME),
new FieldElement("GLTRI", "ACTUAL_FINISH_DT", IDOC.DASHED_DATE),
new FieldElement("GLTRP", "BASIC_FINISH_DT", IDOC.DASHED_DATE),
new FieldElement("GLTRS", "SCHEDULED_FINISH_DT", IDOC.DASHED_DATE),
new FieldElement("GLUZP", "BASIC_FINISH_TM", IDOC.TIME),
new FieldElement("GLUZS", "SCHEDULED_FINISH_TM", IDOC.TIME),
new FieldElement("GMEIN", "COMMON_UNIT_OF_MEASURE_CD"),
new FieldElement("GSTRI", "ACTUAL_START_DT", IDOC.DASHED_DATE),
new FieldElement("GSTRP", "BASIC_START_DT", IDOC.DASHED_DATE),
new FieldElement("GSTRS", "SCHEDULED_START_DT", IDOC.DASHED_DATE),
new FieldElement("GSUZP", "BASIC_START_TM", IDOC.TIME),
new FieldElement("GSUZS", "SCHEDULED_START_TM", IDOC.TIME),
```


SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 251 of 302

[0302] The following shows usage of re-usable formatters defined in the IDOC class:

```
public final static Formatter TIME = new ColonTime();

public final static Formatter DASHED_DATE = new DashedDate();

public final static Formatter STRIP_BLANKS = new RemoveCharFormatter(' ');

public final static Formatter PRODUCT_BASE_NO =
new CompoundFormatter(new Formatter[] {
      New SubStringFormatter("#"),
      STRIP_BLANKS
});

public final static Formatter ITEM_NO = new LeadingZeros(6);

public final static Formatter LINE_NO = new LeadingZeros(4);
```

[0303] Typically SAP instances have some unique aspects of data element handling. Variant document definitions facilitate management of the different aspects by creating a separate document metadata class for each source. Variant documents are recognized based on information in the header of inbound documents. The header information identifies the sending system. Variants can potentially be selected based on other header information. Variant document definitions typically do not supply content metadata. Content metadata of the base document is used. Variant documents are used when the same IDoc has different tag mappings in different circumstances. Typically mappings vary by source SAP instance. Variant document definitions extend the VariantDocumentDefinition class. The super class is called with parameters including selectable variantSelector to identify the variant from information in the header of the inbound IDoc, variantName string to select the variant form of the IDoc in an outbound document, idocSegment containing the format metadata for the variant IDoc, and a string of options. The following illustrates an example of a variant document definition:

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 252 of 302

```
package com.hp.rtsc.documents;
import com.hp.rtsc.metadata.*;
import com.hp.rtsc.metadata.format.*;
import java.sql.SQLException;

// Metadata class to define an ORDERS05 IDoc for Fusion Asia Pacific

public class IhubOrders05FusionAP extends VariantDocumentDefinition {
...
        private static final IDocSegment idocSegment =
        new IDocSegment("ORDERS05", "SALES_ORDER", new Segment[] {
                IDOC.HEADER,
                new Segment("E1EDK01", new FieldElement[] {
...
        // Metadata Constructor:
        public IhubOrders05FusionAP()
                throws MetadataInconsistency, SQLException {
                super(Sources.FusionAP, "ORDERS05FUSIONAP", idocSegment);
        }
}
```

[0304] The following metadata excerpt shows an example of an implementation of source selectors:

```
package com.hp.rtsc.documents;
import com.hp.rtsc.metadata.SourceSelector;

public class Sources
{
    static final SourceSelector Calado = new SourceSelector("SAPCLP");
    static final SourceSelector CRS = new SourceSelector("SAPMA2");
    static final SourceSelector D7 = new SourceSelector("SAPD7C");
    static final SourceSelector EPH = new SourceSelector("SAPEPH");
    static final SourceSelector GPG = new SourceSelector("SAPGPG");
    static final SourceSelector FusionAP = new SourceSelector("SAPAPL");
    static final SourceSelector FusionEU = new SourceSelector("SAPR01");
    static final SourceSelector FusionAM = new SourceSelector("TRMUT");
    static final SourceSelector RSAP = new SourceSelector("SAPPAP");
    static final SourceSelector SAIL = new SourceSelector("SAPCPO");
    static final SourceSelector SPE = new SourceSelector("SAPSPE");
    static final SourceSelector SPR = new SourceSelector("SAPEXE");
}
```

[0305] Aliased documents are used when an IDoc has a distinctive top tag that differentiates the document from other variants of the document. Aliased document definitions extend the AliasedDocumentDefinition class. The super class is called with parameters including idocSegment containing the format metadata for the variant IDoc, and string idocSegment containing the format metadata for the variant IDoc. The following metadata excerpt depicts an example of an aliased document definition:

```
package com.hp.rtsc.documents;
import com.hp.rtsc.metadata.*;
import com.hp.rtsc.metadata.format.*;
import java.sql.SQLException;

// Metadata class to define an ORDERS05 IDoc for Fusion North America

public class IhubOrders05FusionAM extends AliasedDocumentDefinition {
...
        private static final IDocSegment idocSegment =
        new IDocSegment("ZORDREXT", "SALES_ORDER", new Segment[] {
                IDOC.HEADER,
                new Segment("E1EDK01", new TagDefinition[] {
...
        // Metadata Constructor:
        public IhubOrders05FusionAM()
                throws MetadataInconsistency, SQLException {
                super(idocSegment, "ORDERS05");
        }
}
```


SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 255 of 302

[0306] A substitute document definition enables dynamic replacement of an existing base document or aliased document without restarting WebLogic Server (WLS). The substitute document definition can be used to correct or modify document format metadata and deploy the metadata into a system that is executing without interruption. A Java class cannot be reloaded once loaded into a java Virtual Machine (JVM) by the class loader. Therefore, the substitute document definition is created using a new class name. Variant document definitions are substituted in real-time using a new *variant* document definition. The super class is called using parameters including idocSegment that contains format metadata for the variant IDoc. The following metadata excerpt depicts an example of a substitute document definition:

```
package com.hp.rtsc.documents;
import com.hp.rtsc.metadata.*;
import com.hp.rtsc.metadata.format.*;
import java.sql.SQLException;

// Metadata class to define an ORDERS05 IDoc

public class Orders05v3 extends SubstituteDocumentDefinition {
...
        private static final IDocSegment idocSegment =
        new IDocSegment("ORDERS05", "SALES_ORDER", new Segment[] {
                IDOC.HEADER,
                new Segment("E1EDK01", new FieldElement[] {
...
        // Metadata Constructor:
        public IhubOrders05()
                throws MetadataInconsistency, SQLException {
                super(idocSegment);
        }
}
```

[0307] The same SQL statements generated for base document definitions can be used for variant, aliased, and substitute documents, saving SQL statement compilation time, conserving SQL connections, and enabling format metadata to be deployed without recompilation of SQL statements. A management interface notifies the running system when a new document definition class is available, whether a replacement or completely new document definition.

[0308] While the present disclosure describes various embodiments, these embodiments are to be understood as illustrative and do not limit the claim scope. Many variations, modifications, additions and improvements of the described embodiments are possible. For example, those having ordinary skill in the art will readily implement the steps necessary to provide the structures and methods disclosed herein, and will understand that the process parameters, materials, and dimensions are given by way of example only. The parameters, materials, components, and dimensions can be varied to achieve the desired structure as well as modifications, which are within the scope of the claims. Variations and modifications of the embodiments disclosed herein may also be made while remaining within the scope of the following claims. For example, the specific embodiments described herein identify various computing languages, application software packages, and systems that operate as part of or in conjunction with the illustrative information handling system, components, and methods. In other embodiments, any suitable languages, applications, systems, and operating methods may be used. The various embodiments described herein have multiple aspects and components. These aspects and components may be implemented individually or in combination in various embodiments and applications. Accordingly, each claim is to be considered individually and not to include aspects or limitations that are outside the wording of the claim.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 257 of 302

WHAT IS CLAIMED IS:

1. A data services handler comprising:

an interface between a data store and applications that supply and consume data; and

a real time information director (RTID) that transforms data under direction of polymorphic metadata that defines a security model and data integrity rules for application to the data.

2. The data services handler according to Claim 1 further comprising:

a data mapper capable of collecting data from a plurality of diverse applications that can supply data in multiple diverse external document formats and mapping the supplied data to a predetermined unified data model according to the polymorphic metadata.

3. The data services handler according to Claim 1 wherein:

the real time information director comprises a logic that cleanses and enriches the data, supplies data security, and insures data integrity.

4. The data services handler according to Claim 1 wherein the real time information director further comprises:

a disassembler that disassembles inbound data documents into multiple records in preparation for insertion into metadata tables; and

an assembler that assembles outbound data documents from at least one record selected from at least one metadata-selected table.

5. The data services handler according to Claim 1 wherein the real time information director further comprises:

a logic that defines a data model according to received descriptive metadata that identifies individual data elements and specifies logic behaviors applicable to the individual data elements.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 258 of 302

6. The data services handler according to Claim 1 wherein the real time information director further comprises:

a logic that represents metadata as polymorphic objects, enabling individual metadata objects to have distinct numbers and types of attributes according to a predetermined object class, the behaviors and attributes being defined specific to object class.

7. The data services handler according to Claim 1 wherein the real time information director further comprises:

a logic that represents metadata as stateless polymorphic objects that are permanent after initial construction, whereby methods of stateful metadata classes delegate operation to the stateless polymorphic metadata objects.

8. The data services handler according to Claim 1 wherein the real time information director further comprises:

at least one metadata object constructed using constructors of multiple metadata classes, the constructors having parameters that specify metadata data elements and being nested to represent hierarchical relationships between the specified metadata data elements, the at least one metadata object including identifying information for deriving the hierarchical relationships.

9. The data services handler according to Claim 1 wherein the real time information director further comprises:

a logic that uses received data document mapping metadata to disassemble elements of inbound data documents into rows and columns to be inserted into inbound metadata-selected tables in a database, select rows and columns from outbound metadata-selected tables of the database and assemble elements from the outbound metadata-selected tables into an outbound data document, and generate a schema for documents that can be input or output using the received data document mapping metadata.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 259 of 302

10. The data services handler according to Claim 1 wherein:

the real time information director receives and publishes data documents having a content and a format defined by document metadata.

11. The data services handler according to Claim 1 wherein the real time information director further comprises:

a logic that reads a schema of tables identified in metadata; and

a logic that automatically generates and caches prepare SQL statements from a content metadata portion of document metadata according to the schema.

12. The data services handler according to Claim 1 further comprising:

a servlet that invokes the real time information director synchronously when an inbound data document is delivered via HyperText Transport Protocol (HTTP);

a logic that invokes the real time information director asynchronously when an inbound data document is delivered via Java Message Service (JMS);

a servlet that invokes the real time information director synchronously when an outbound data document is requested via HyperText Transport Protocol (HTTP); and

a logic that invokes the real time information director asynchronously when an outbound data document is requested via Java Message Service (JMS).

13. The data services handler according to Claim 1 wherein the real time information director further comprises:

a subscription handler that responds to a triggering event by publishing a document according to content and format specified by Document Metadata, and sending the document to a designated queue or topic.

14. The data services handler according to Claim 1 wherein the real time information director further comprises:

a security manager that identifies an agent on behalf of whom information is queried or updated in a document request, dynamically assigns an internal unique identifier to consumers and agents upon registration; and

the real time information director grants access to individual documents based on security policies according to a relationship between an agent, a consumer affected by the agent, and permissions granted by the consumer.

15. The data services handler according to Claim 14 wherein:
the security manager enforces securities policies that are specifiable for each document.

16 The data services handler according to Claim 14 wherein:
the security manager enforces securities policies that are customized to identify specific agent and consumer relationships expressed in a business model.

17. The data services handler according to Claim 1 wherein the real time information director further comprises:
an audit log handler that responds to changes in the data store by compiling and storing information based on metadata that determines content and detail of the compiled information.

18. The data services handler according to Claim 1 wherein the real time information director further comprises:
a data cleanser that receives data elements identified by Document Metadata, normalizes the data elements, queues source and destination information for detection of duplicates, and cross-references duplicates in the data store.

19. The data services handler according to Claim 1 wherein the real time information director further comprises:
an aggregation engine that computes aggregates in memory from data selected from the data store.

20. The data services handler according to Claim 1 wherein the real time information director further comprises:
a real-time scoring engine.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 261 of 302

21. The data services handler according to Claim 1 wherein the real time information director further comprises:

a business rules engine that defines a business object model capable of enabling access to data described by Document Metadata, performing computations on data relevant to business rules specified by the business object model, concluding a decision based on computation results, and executing the decision.

22. An article of manufacture comprising:

a controller-usable medium having a computable readable program code embodied therein for managing data services, the computable readable program code further comprising:

a code capable of causing the controller to transfer data among a data store and applications that supply and consume data; and

a code capable of causing the controller to transform data for transfer using a representation of metadata as polymorphic objects, enabling individual metadata objects to have distinct numbers and types of attributes according to a predetermined object class, the behaviors and attributes being defined specific to object class.

23. The article of manufacture according to Claim 22 further comprising:

a code capable of causing the controller to transform data under direction of polymorphic metadata that defines a security model and data integrity rules for application to the data.

24. The article of manufacture according to Claim 22 further comprising:

a code capable of causing the controller to collect data from a plurality of diverse applications that can supply data in multiple diverse external document formats and mapping the supplied data to a predetermined unified data model according to the polymorphic metadata.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 262 of 302

25. The article of manufacture according to Claim 22 further comprising:

a code capable of causing the controller to represent metadata as stateless polymorphic objects that are permanent after initial construction, whereby methods of stateful metadata classes delegate operation to the stateless polymorphic metadata objects.

26. The article of manufacture according to Claim 22 further comprising:

a code capable of causing the controller to construct at least one metadata object using constructors of multiple metadata classes;

a code capable of causing the controller to use constructor parameters to specify metadata data elements and being nested to represent hierarchical relationships between the specified metadata data elements, the at least one metadata object including identifying information for deriving the hierarchical relationships.

27. The article of manufacture according to Claim 22 further comprising:

a code capable of causing the controller to use received data document mapping metadata to disassemble elements of inbound data documents into rows and columns to be inserted into inbound metadata-selected tables in a database, select rows and columns from outbound metadata-selected tables of the database and assemble elements from the outbound metadata-selected tables into an outbound data document, and generate a schema for documents that can be input or output using the received data document mapping metadata.

28. An enterprise system comprising:

a real time information director (RTID) that manages information flow between a data store and applications that supply and consume data, including transforming data under direction of polymorphic metadata.

29. The enterprise system according to Claim 28 further comprising:
a business rules engine coupled to the real time information director that personalizes transactions by applying business rules and prediction models against cached entity data.

30. The enterprise system according to Claim 28 further comprising:
a server platform coupled to the real time information director that serves deployed models, customer data, and interaction data.

31. The enterprise system according to Claim 28 wherein:
the RTID represents metadata as polymorphic objects, enabling individual metadata objects to have distinct numbers and types of attributes according to a predetermined object class, the behaviors and attributes being defined specific to object class.

32. The enterprise system according to Claim 28 wherein:
the RTID represents metadata as stateless polymorphic objects that are permanent after initial construction, whereby methods of stateful metadata classes delegate operation to the stateless polymorphic metadata objects.

33. A data services handler comprising:
a real time information director (RTID) that maps documents into a data model further comprising:
a disassembly logic that disassembles documents for loading into a database;
an assembly logic that assembles documents from records in the database; and
a generator logic that generates an eXtended Markup Language (XML) Schema that describes format of the documents,
the disassembly, assembly, and generator logic being operative using a single descriptive polymorphic metadata.

34. The data services handler according to Claim 33 wherein the metadata further comprises:

content metadata that defines data to be included into a document; and

format metadata that defines how the data appears in the document.

35. The data services handler according to Claim 33 wherein the real time information director further comprises:

a query logic that queries the database to determine information about tables in the database that the document uses, the information including field name, field datatype, and field allowance of nulls.

36. A method for generating native documents as an eXtended Markup Language (XML) representation of data in a Structured Query Language (SQL) database, the method comprising:

creating metadata for a new native document further comprising:

determining whether inbound documents are inserted, delserted or upserted;

determining what data from the SQL database is sufficient for usage in the new native document; and

creating an XML Schema for the native document.

37. A content management system comprising:

content metadata classes that describe a relationship between records and tables contributing to a document in an information handling system;

a Structured Query Language (SQL) schema that corresponds to the content metadata and includes record definitions and field definitions;

a database arranged as native documents in an eXtended Markup Language (XML) representation; and

SQL connections and prepared statements including document handlers, the database, document handlers, and metadata being arranged in parallel trees.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 265 of 302

38. An article of manufacture comprising:

a controller-usable medium having a computable readable program code embodied therein for generating native documents as an eXtended Markup Language (XML) representation of data in a Structured Query Language (SQL) database, the computable readable program code further comprising:

a code capable of causing the controller to create metadata for a new native document;

a code capable of causing the controller to determine whether inbound documents are delserted or upserted;

a code capable of causing the controller to determine what data from the SQL database is sufficient for usage in the new native document; and

a code capable of causing the controller to create an XML Schema for the native document.

39. A data handling system comprising:

means for generating native documents as an eXtended Markup Language (XML) representation of data in a Structured Query Language (SQL) database;

means for creating metadata for a new native document;

means for determining whether inbound documents are delserted or upserted;

means for determining what data from the SQL database is sufficient for usage in the new native document; and

means for creating an XML Schema for the native document.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 266 of 302

DATA SERVICES HANDLER

Steven R. Carr
Venkat Muthuswamy
Tudor Har

ABSTRACT

A data services handler comprises an interface between a data store and applications that supply and consume data, and a real time information director (RTID) that transforms data under direction of polymorphic metadata that defines a security model and data integrity rules for application to the data.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 267 of 302

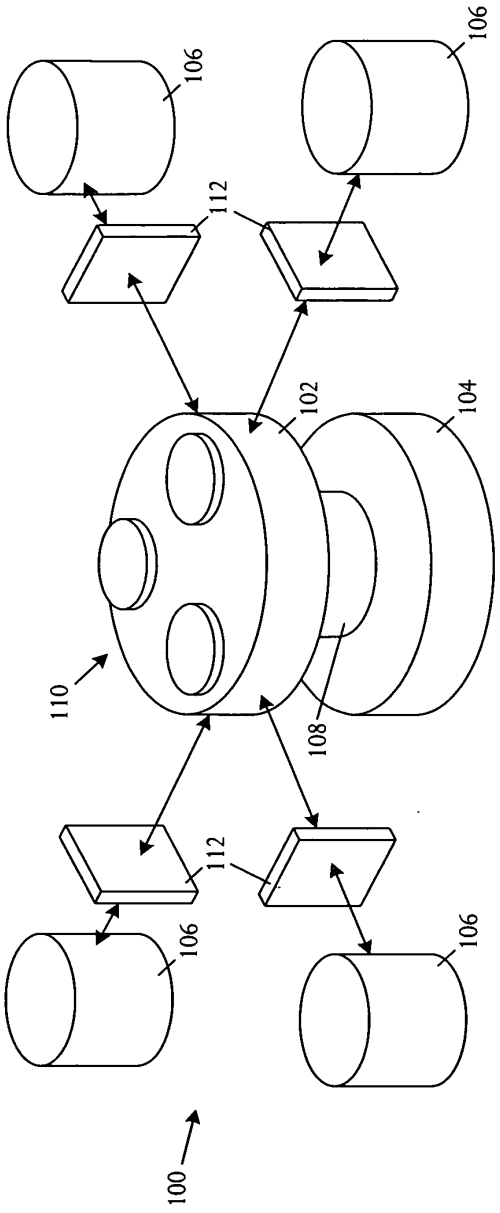


FIG. 1A

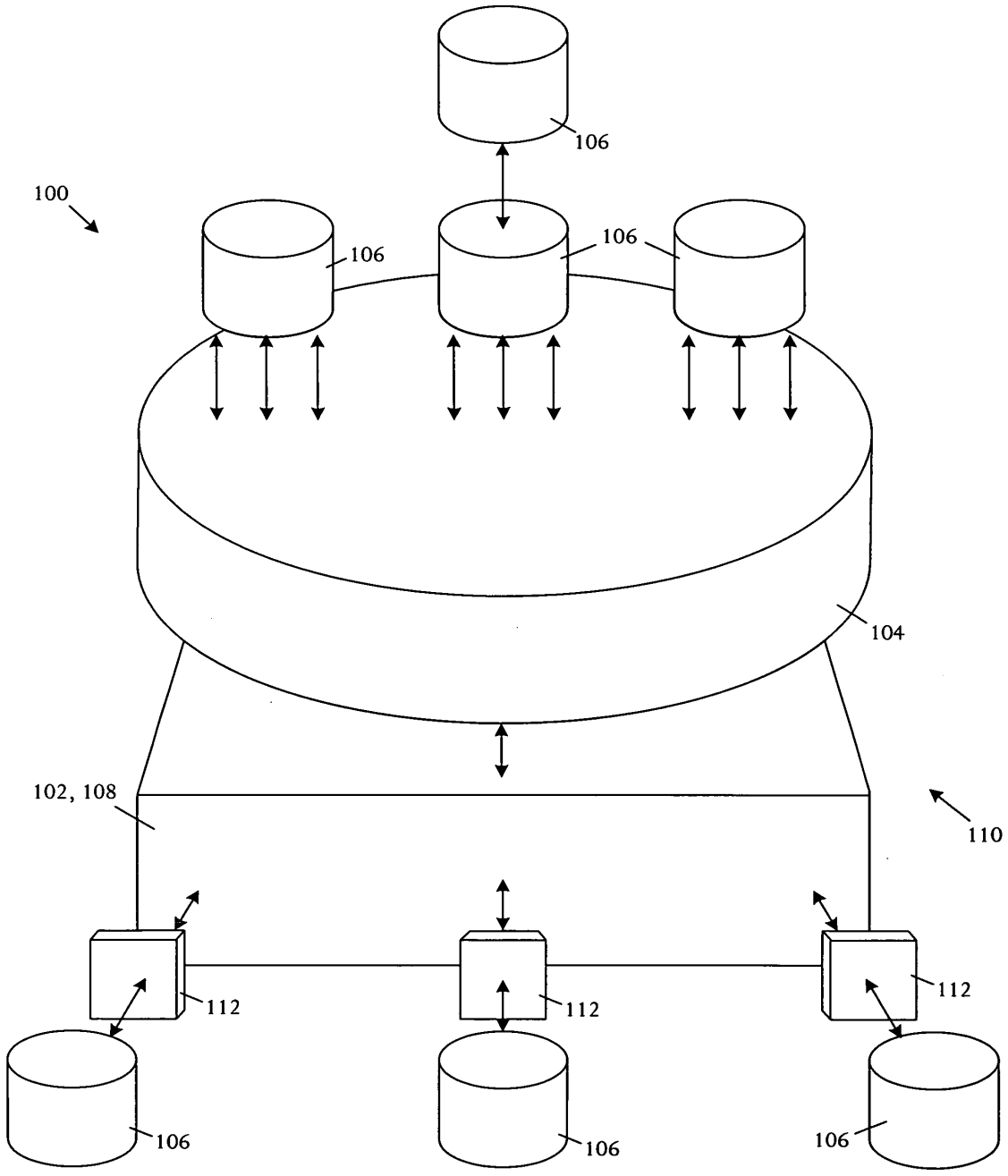


FIG. 1B

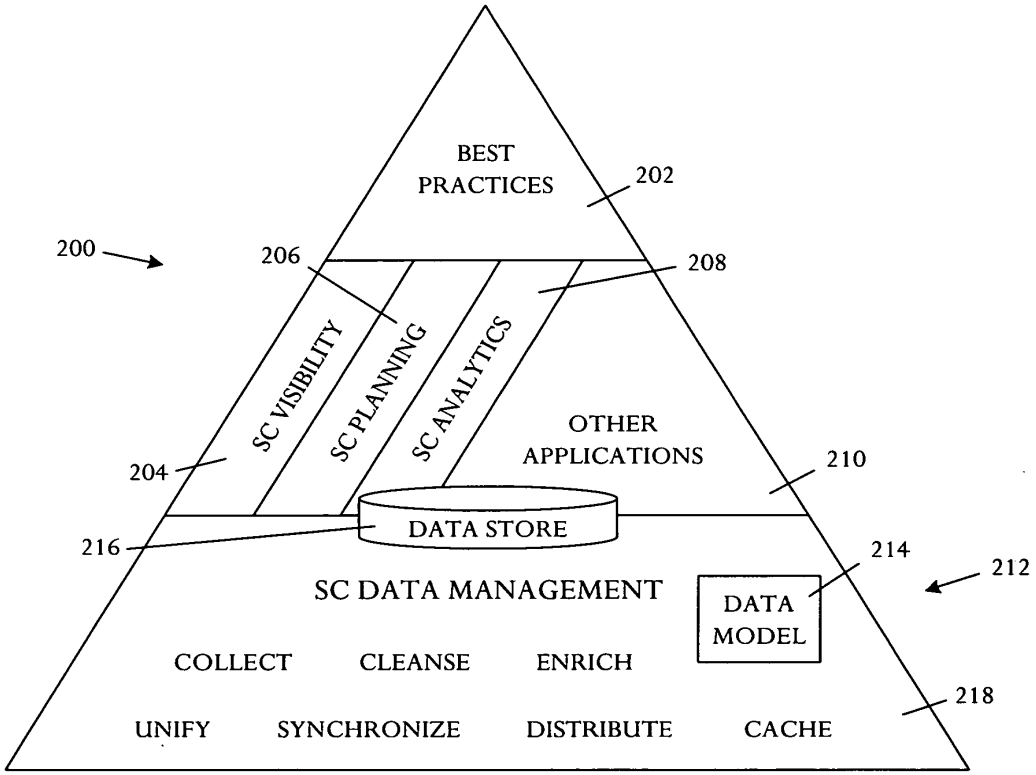

BEST PRACTICES
202
200
206
208
SC VISIBILITY
SC PLANNING
SC ANALYTICS
OTHER APPLICATIONS
204
210
DATA STORE
216
214
212
SC DATA MANAGEMENT
DATA MODEL
COLLECT
CLEANSE
ENRICH
218
UNIFY
SYNCHRONIZE
DISTRIBUTE
CACHE

FIG. 2A

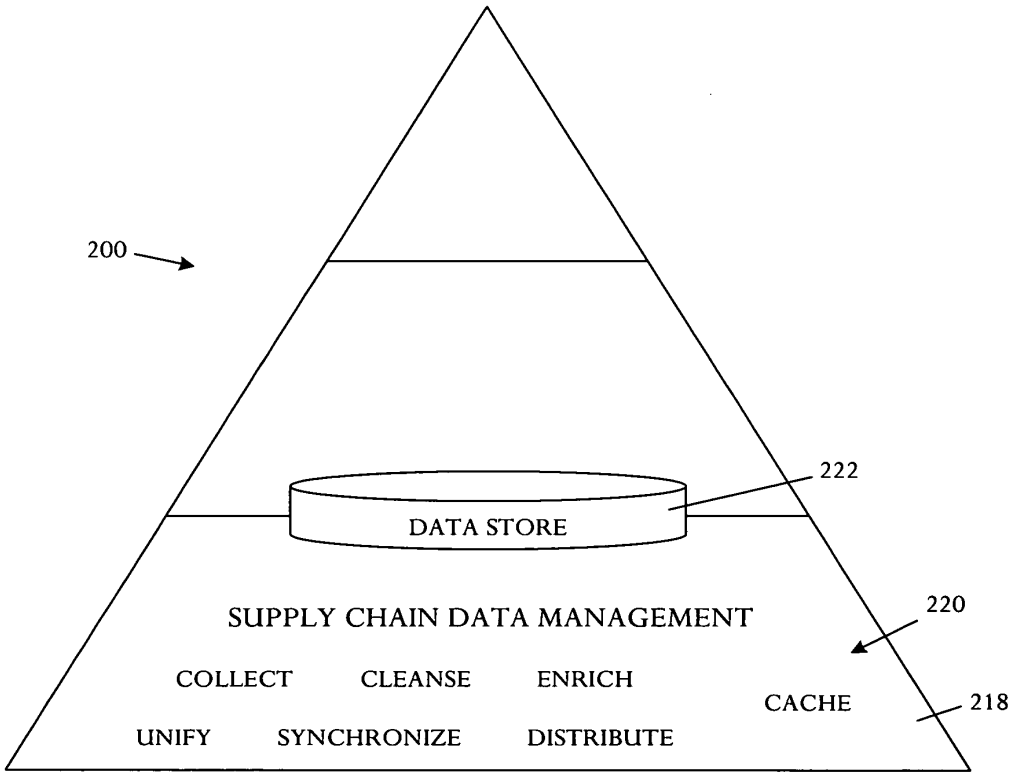

200
222
DATA STORE
SUPPLY CHAIN DATA MANAGEMENT
220
COLLECT
CLEANSE
ENRICH
CACHE
218
UNIFY
SYNCHRONIZE
DISTRIBUTE

FIG. 2B

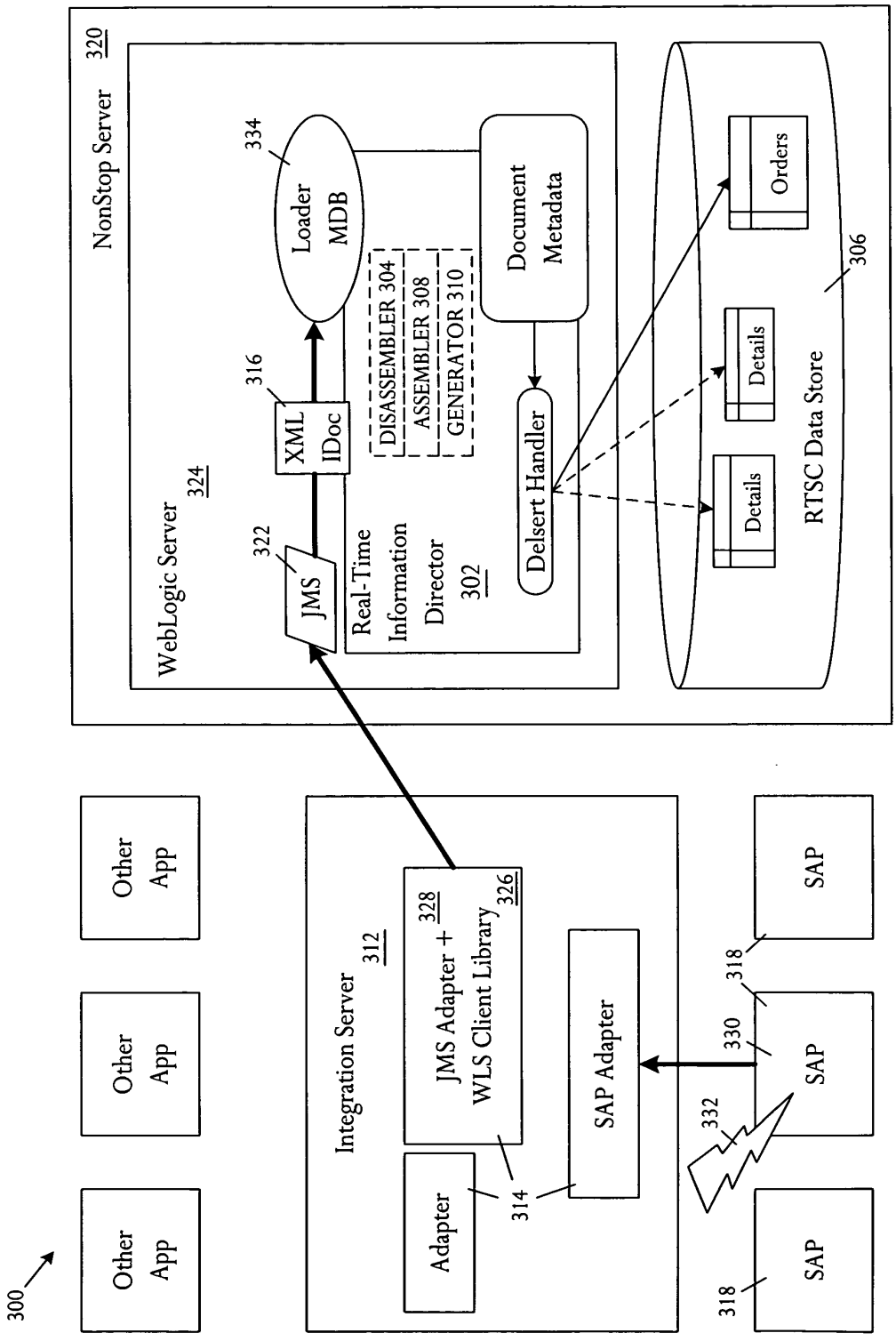

300
Other App
Other App
Other App
Integration Server
312
Adapter
JMS Adapter + WLS Client Library
328
326
314
SAP Adapter
SAP
318
330
SAP
318
332
SAP
NonStop Server
320
WebLogic Server
324
JMS
322
XML IDoc
316
Loader MDB
334
Real-Time Information Director
302
DISASSEMBLER 304
ASSEMBLER 308
GENERATOR 310
Document Metadata
Delsert Handler
RTSC Data Store
306
Orders
Details
Details

FIG. 3A

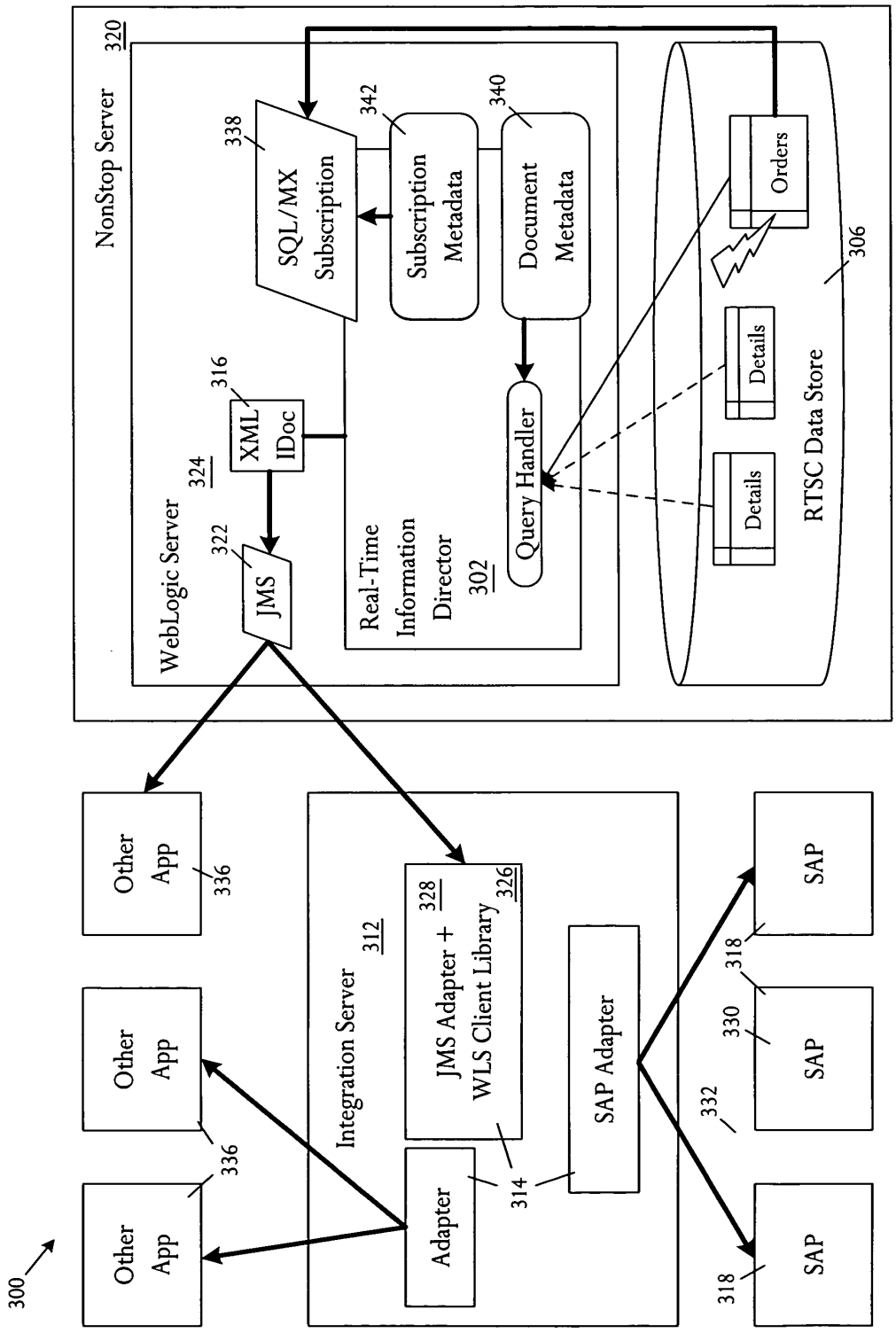

300
NonStop Server
320
WebLogic Server
324
Real-Time Information Director
302
SQL/MX Subscription
338
Subscription Metadata
342
Document Metadata
340
Query Handler
XML IDoc
316
JMS
322
Orders
Details
Details
RTSC Data Store
306
Other App
336
Other App
336
Other App
Integration Server
312
JMS Adapter + WLS Client Library
328
326
Adapter
314
SAP Adapter
SAP
318
SAP
330
SAP
318
332

FIG. 3B

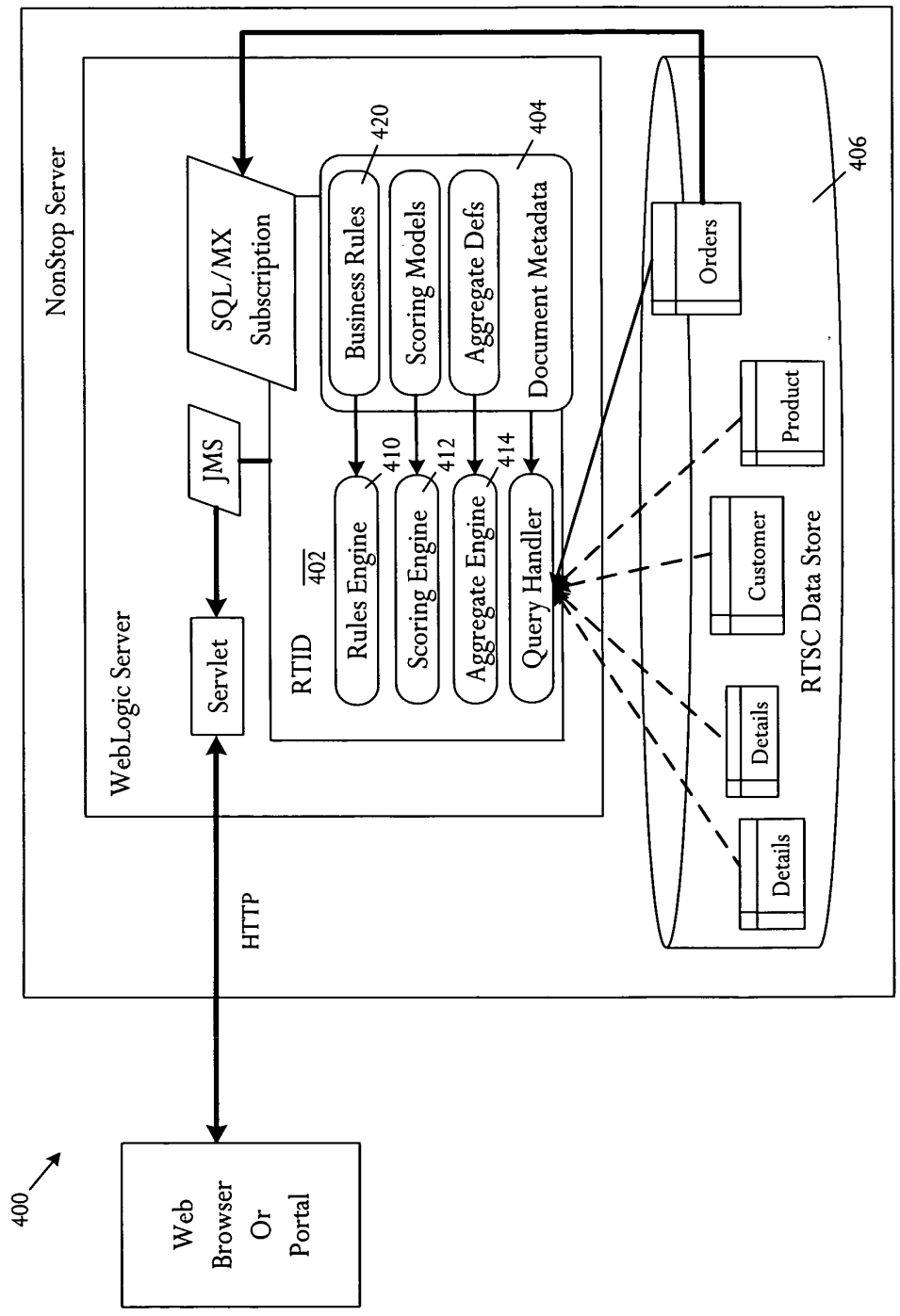

NonStop Server
WebLogic Server
SQL/MX Subscription
JMS
Servlet
HTTP
RTID
402
420
Business Rules
Scoring Models
Aggregate Defs
Document Metadata
404
Rules Engine
Scoring Engine
Aggregate Engine
Query Handler
410
412
414
406
Orders
Product
Customer
Details
Details
RTSC Data Store
400
Web Browser Or Portal

FIG. 4

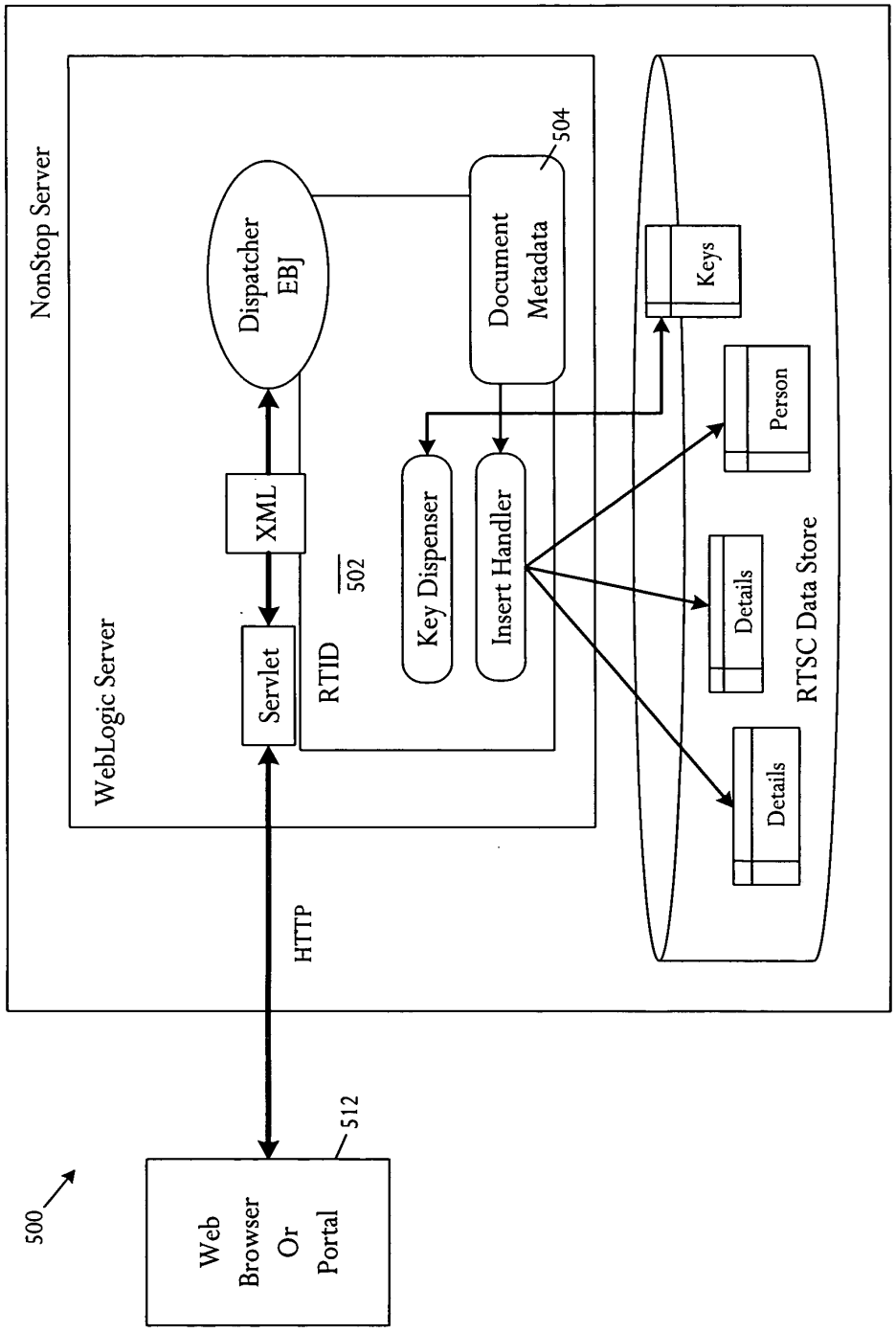

500
Web Browser Or Portal
512
HTTP
NonStop Server
WebLogic Server
Servlet
XML
Dispatcher EBJ
RTID
502
Key Dispenser
Insert Handler
Document Metadata
504
RTSC Data Store
Keys
Person
Details
Details

FIG. 5A

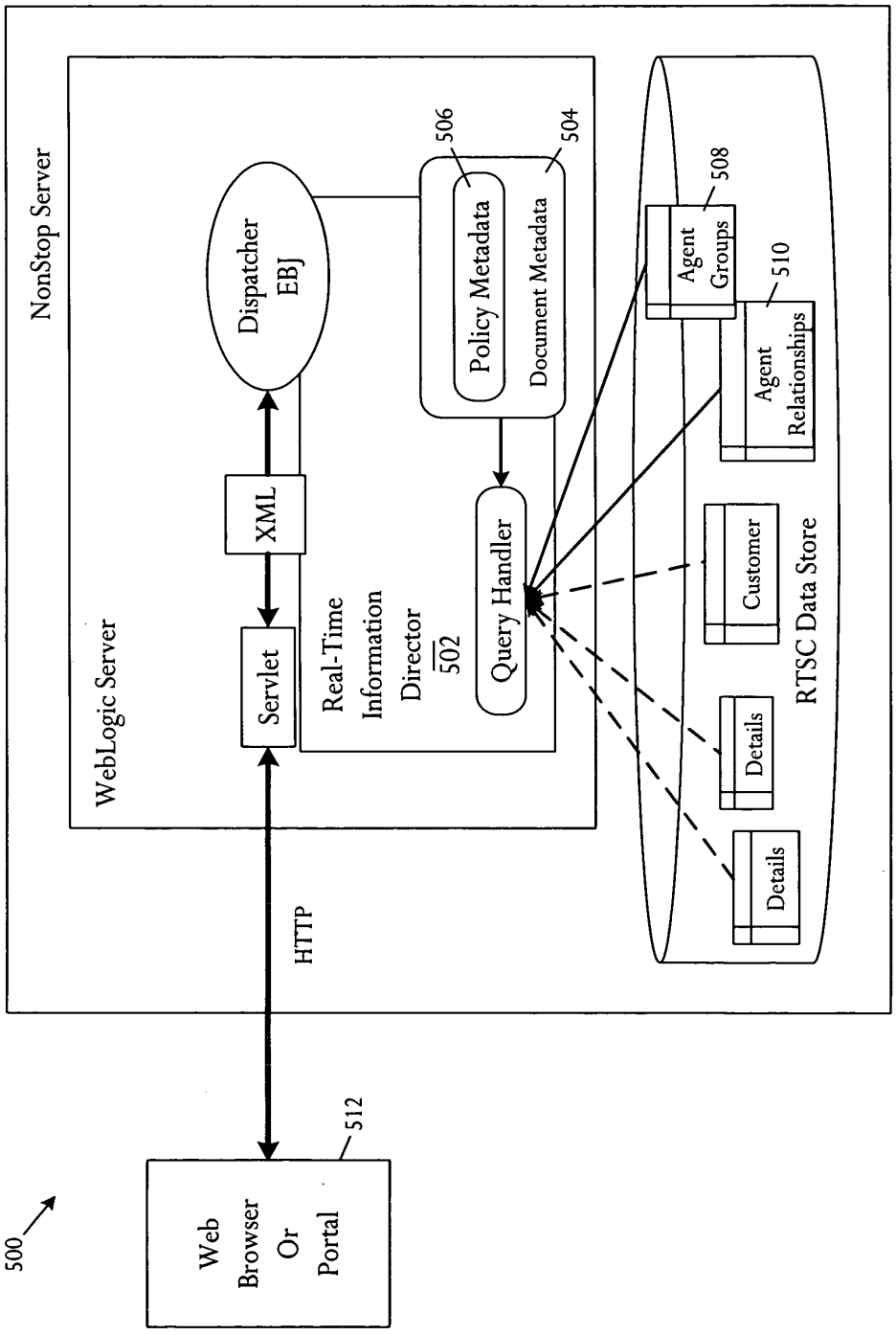

500
Web Browser Or Portal
512
HTTP
NonStop Server
WebLogic Server
Dispatcher EBJ
XML
Servlet
Real-Time Information Director 502
Query Handler
506
Policy Metadata
504
Document Metadata
508
Agent Groups
510
Agent Relationships
Customer
Details
Details
RTSC Data Store

FIG. 5B

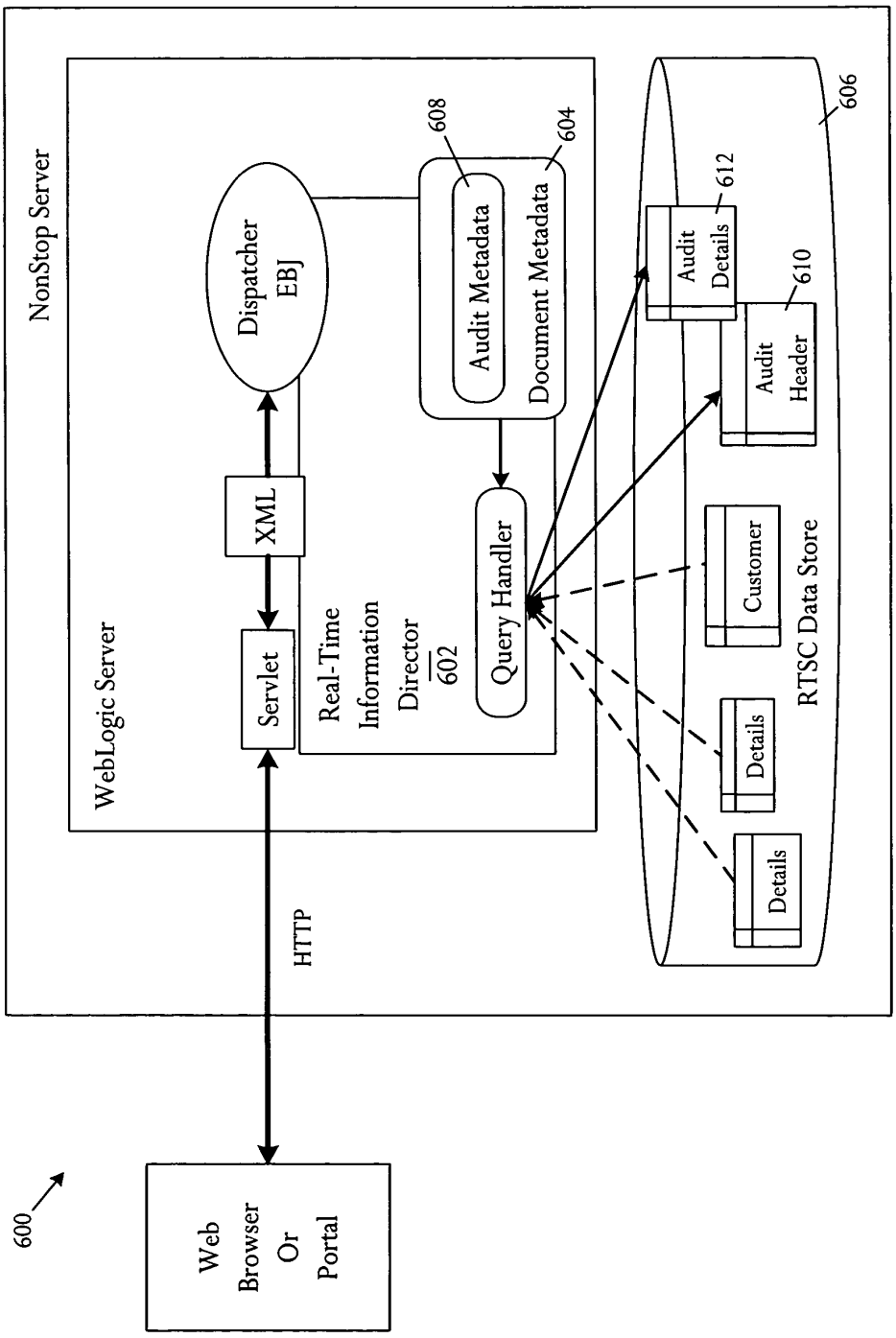

600
Web Browser Or Portal
HTTP
NonStop Server
WebLogic Server
Servlet
XML
Dispatcher EBJ
Real-Time Information Director 602
Query Handler
608
Audit Metadata
Document Metadata
604
606
RTSC Data Store
Audit Details
612
Audit Header
610
Customer
Details
Details

FIG. 6

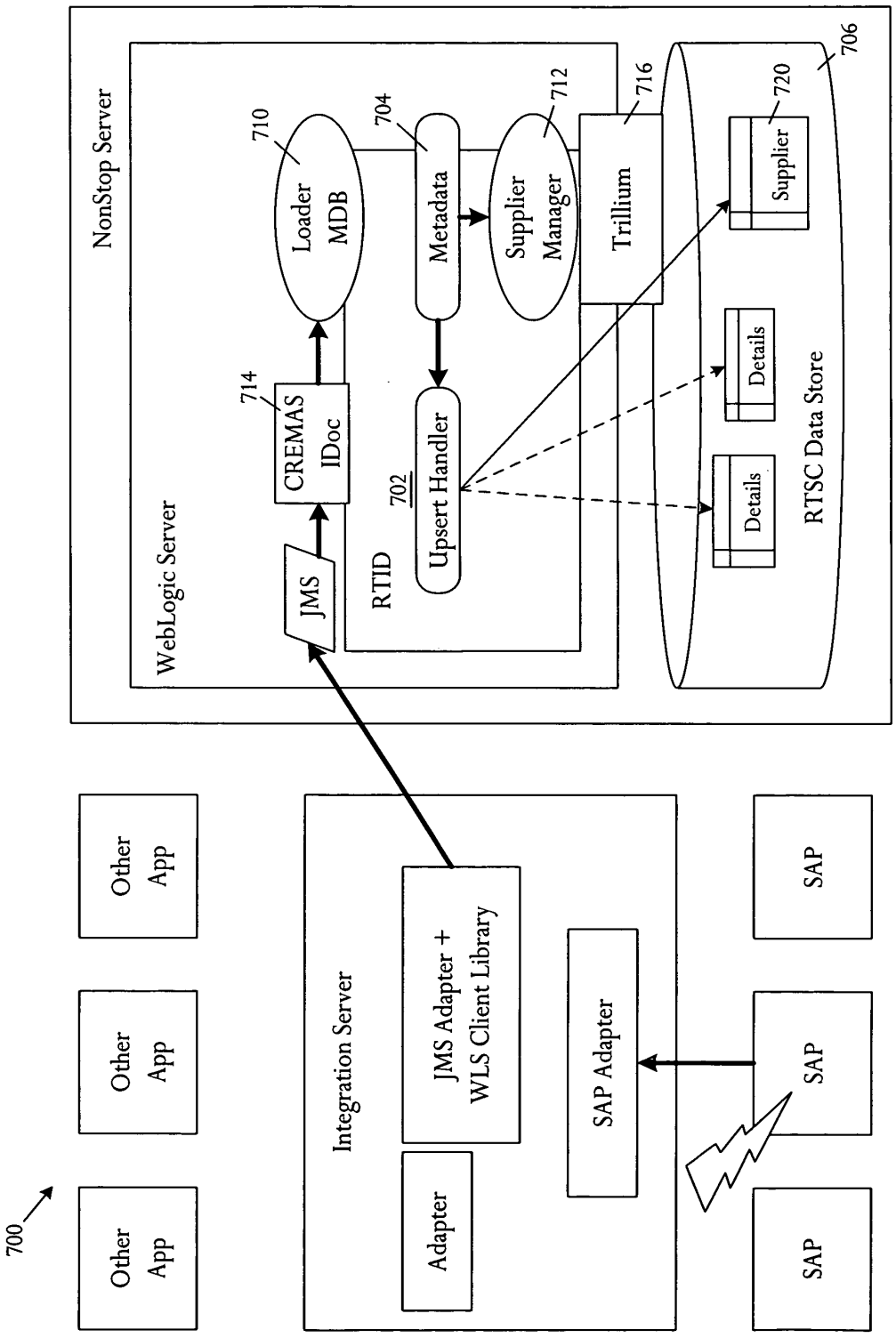

700
Other App
Other App
Other App
Integration Server
Adapter
JMS Adapter + WLS Client Library
SAP Adapter
SAP
SAP
NonStop Server
WebLogic Server
JMS
CREMAS IDoc
714
Loader MDB
710
RTID
Upsert Handler
702
Metadata
704
Supplier Manager
712
Trillium
716
Details
Details
Supplier
720
RTSC Data Store
706

FIG. 7A

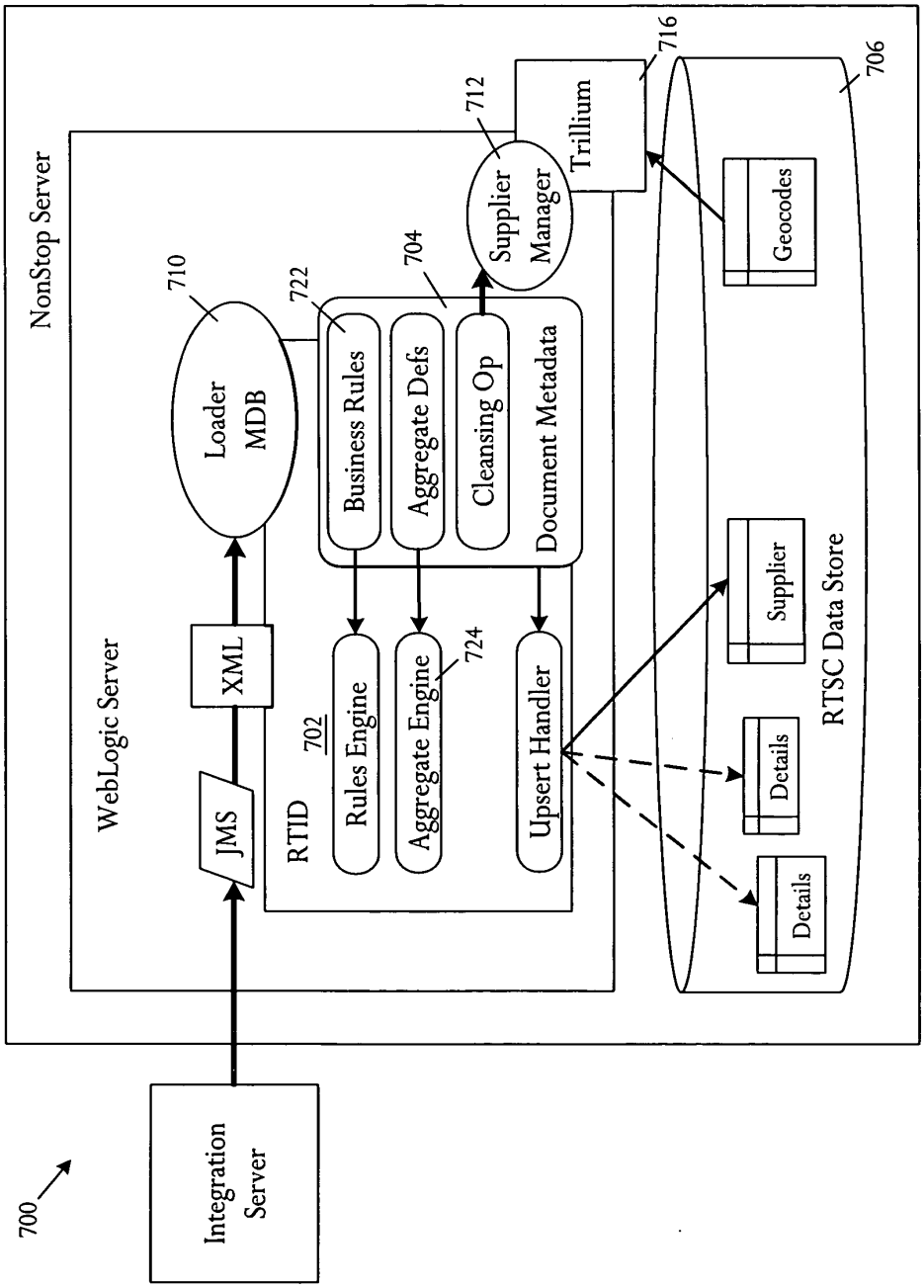

700
Integration Server
NonStop Server
WebLogic Server
JMS
XML
Loader MDB
710
RTID
702
Rules Engine
Aggregate Engine
724
Upsert Handler
722
Business Rules
Aggregate Defs
Cleansing Op
Document Metadata
704
Supplier Manager
712
Trillium
716
706
Geocodes
Supplier
Details
Details
RTSC Data Store

FIG. 7B

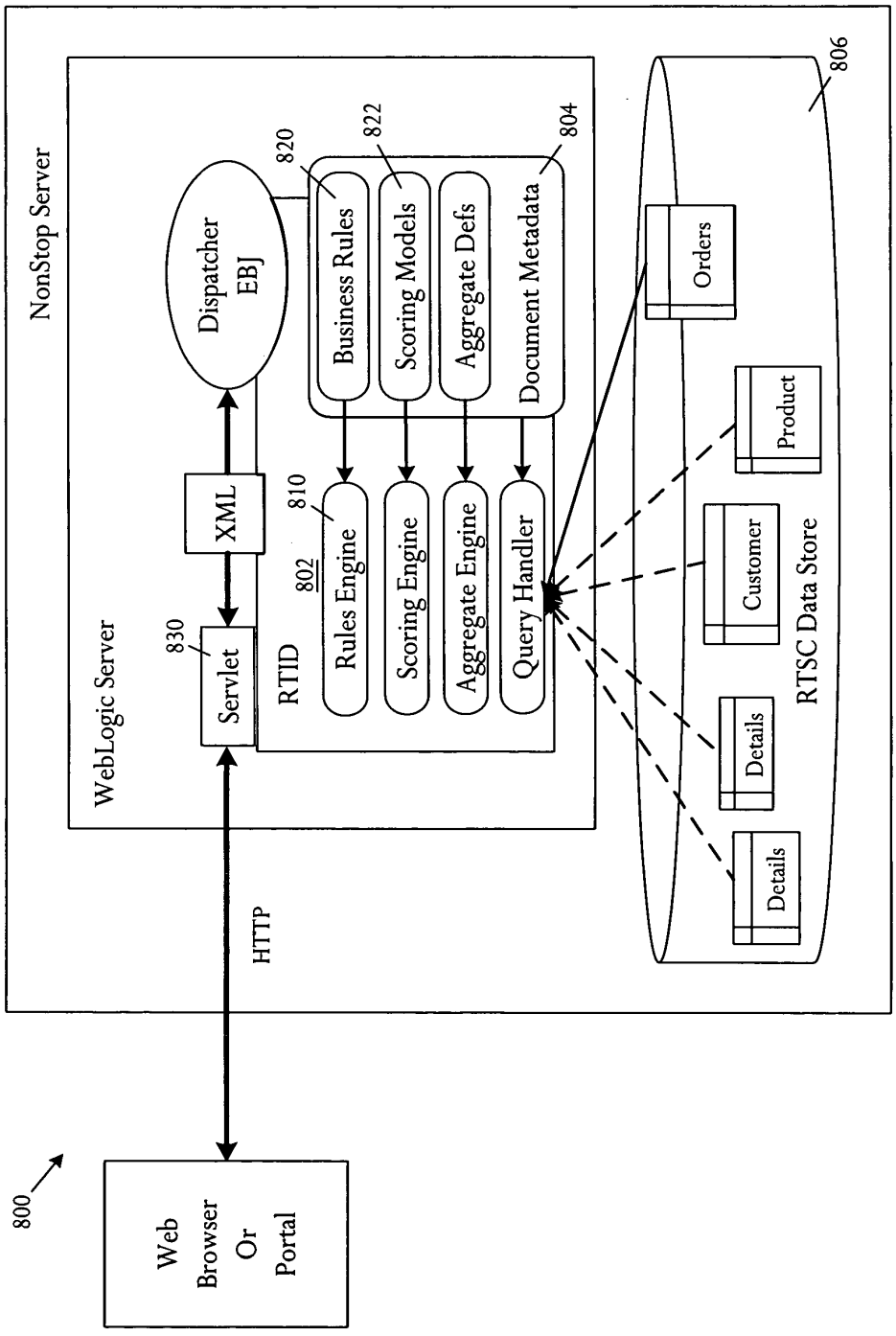

800
Web Browser Or Portal
HTTP
NonStop Server
WebLogic Server
Dispatcher EBJ
XML
Servlet
830
RTID
802
Rules Engine
Scoring Engine
Aggregate Engine
Query Handler
810
820
Business Rules
Scoring Models
Aggregate Defs
Document Metadata
822
804
806
Orders
Product
Customer
Details
Details
RTSC Data Store

FIG. 8

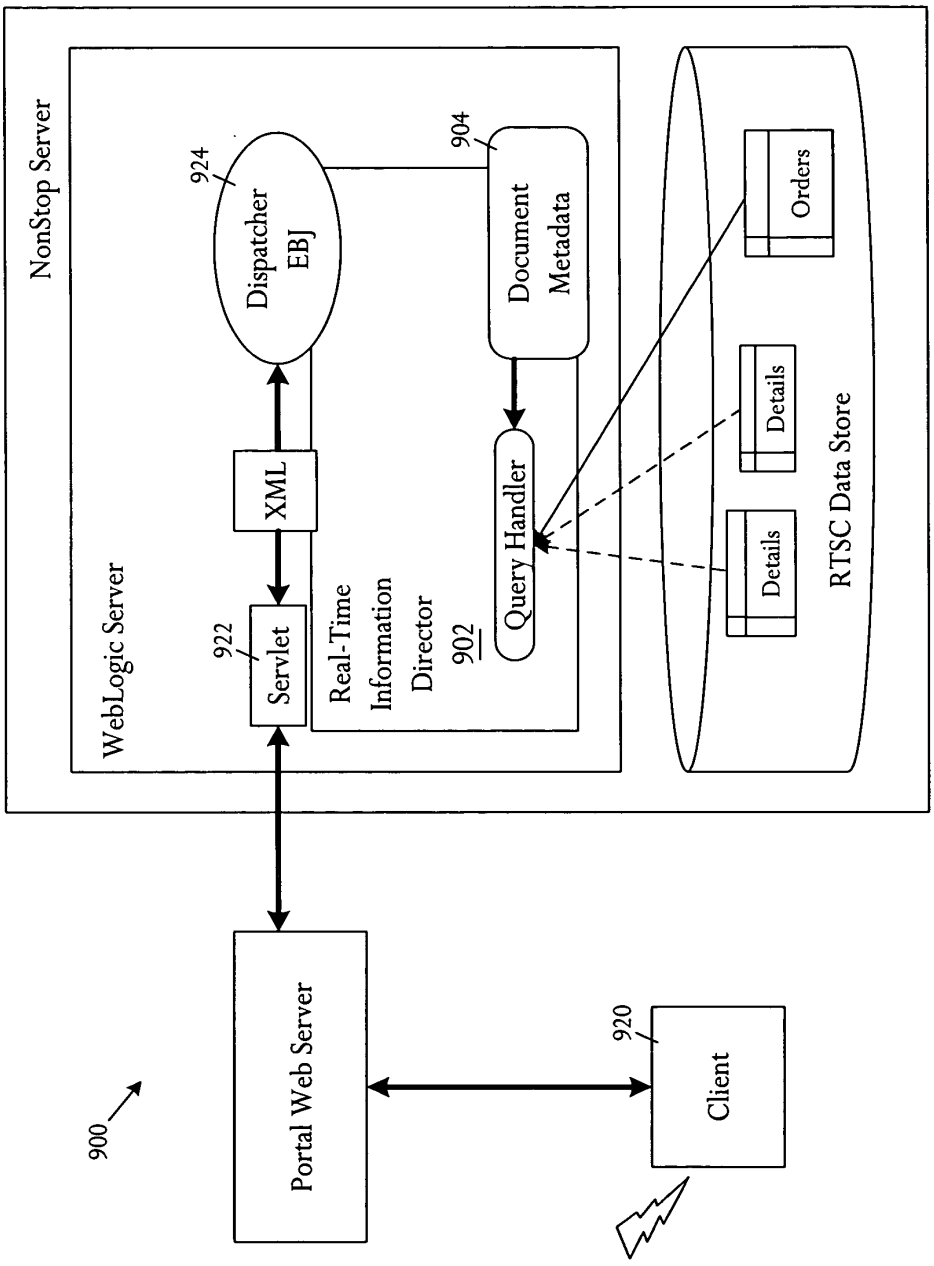

NonStop Server
WebLogic Server
924
Dispatcher EBJ
904
Document Metadata
XML
922
Servlet
Real-Time Information Director
902
Query Handler
Orders
Details
Details
RTSC Data Store
900
Portal Web Server
920
Client

FIG. 9

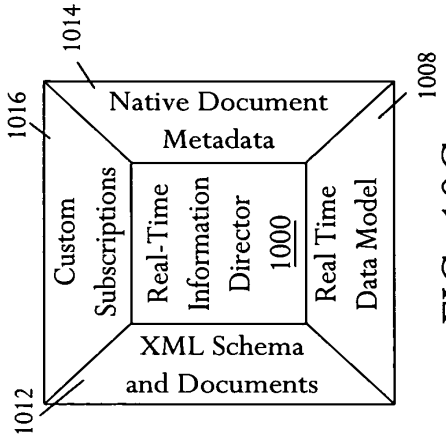


FIG. 10C

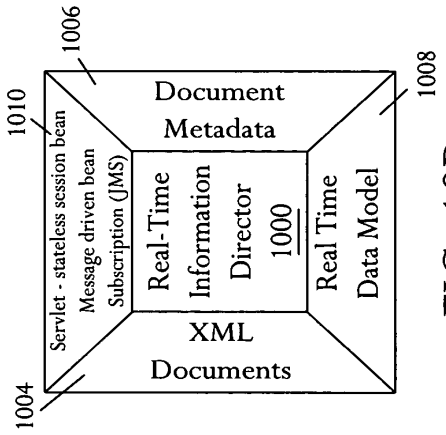


FIG. 10B

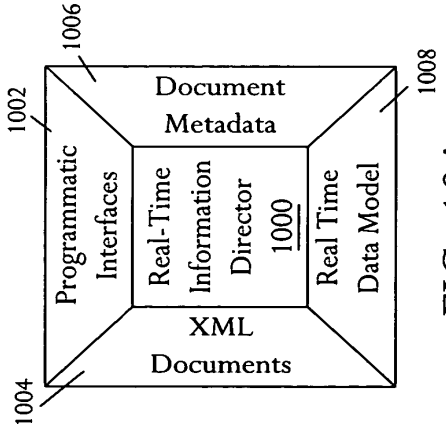


FIG. 10A

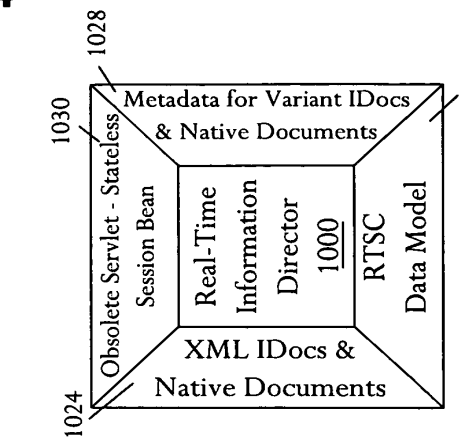


FIG. 10F

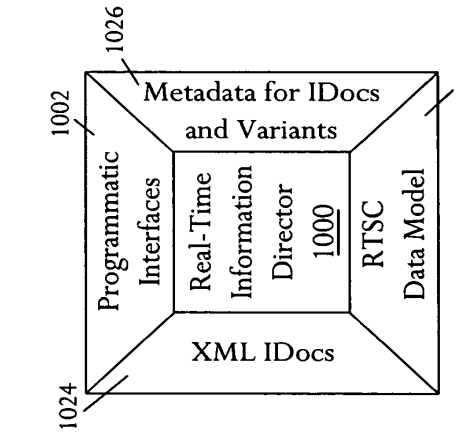


FIG. 10E

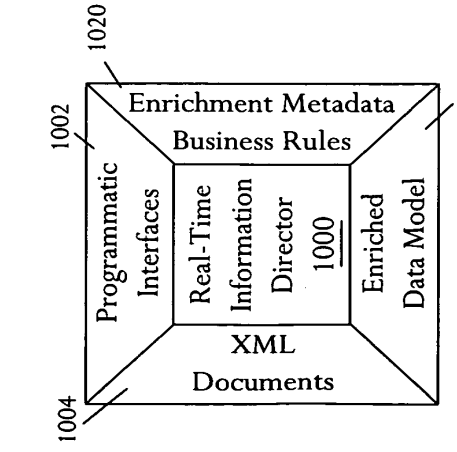


FIG. 10D

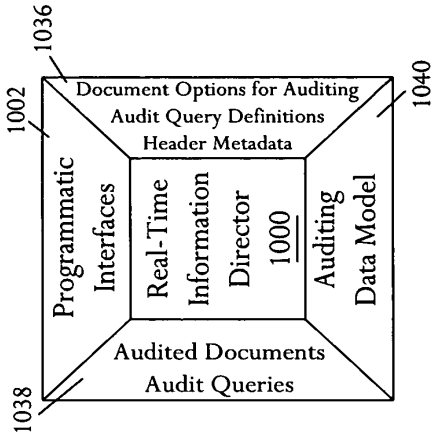


FIG. 10I

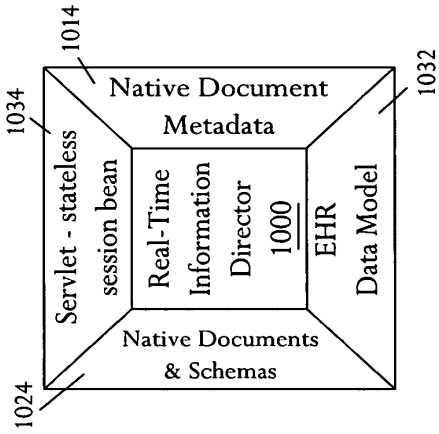


FIG. 10H

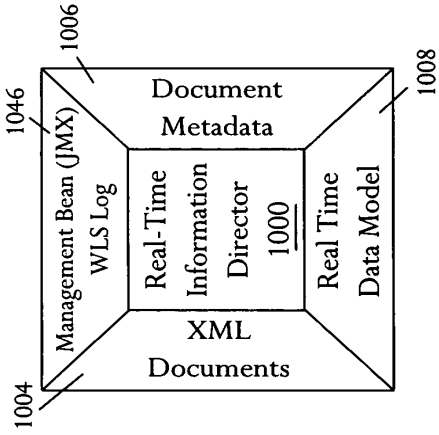


FIG. 10K

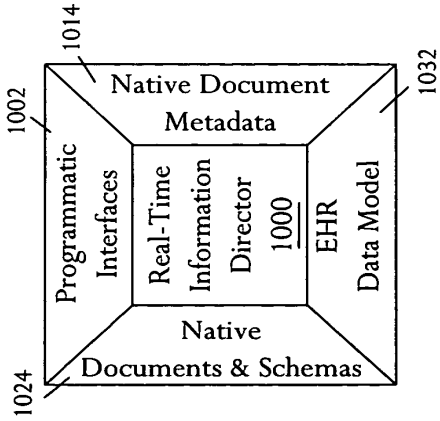


FIG. 10G

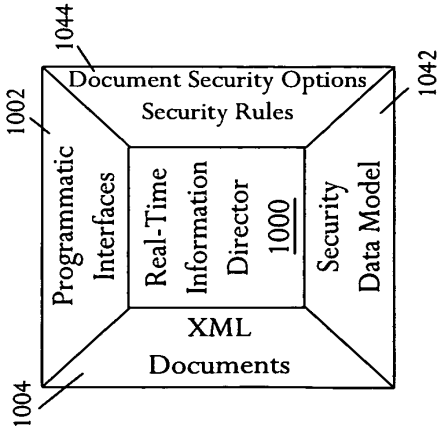


FIG. 10J

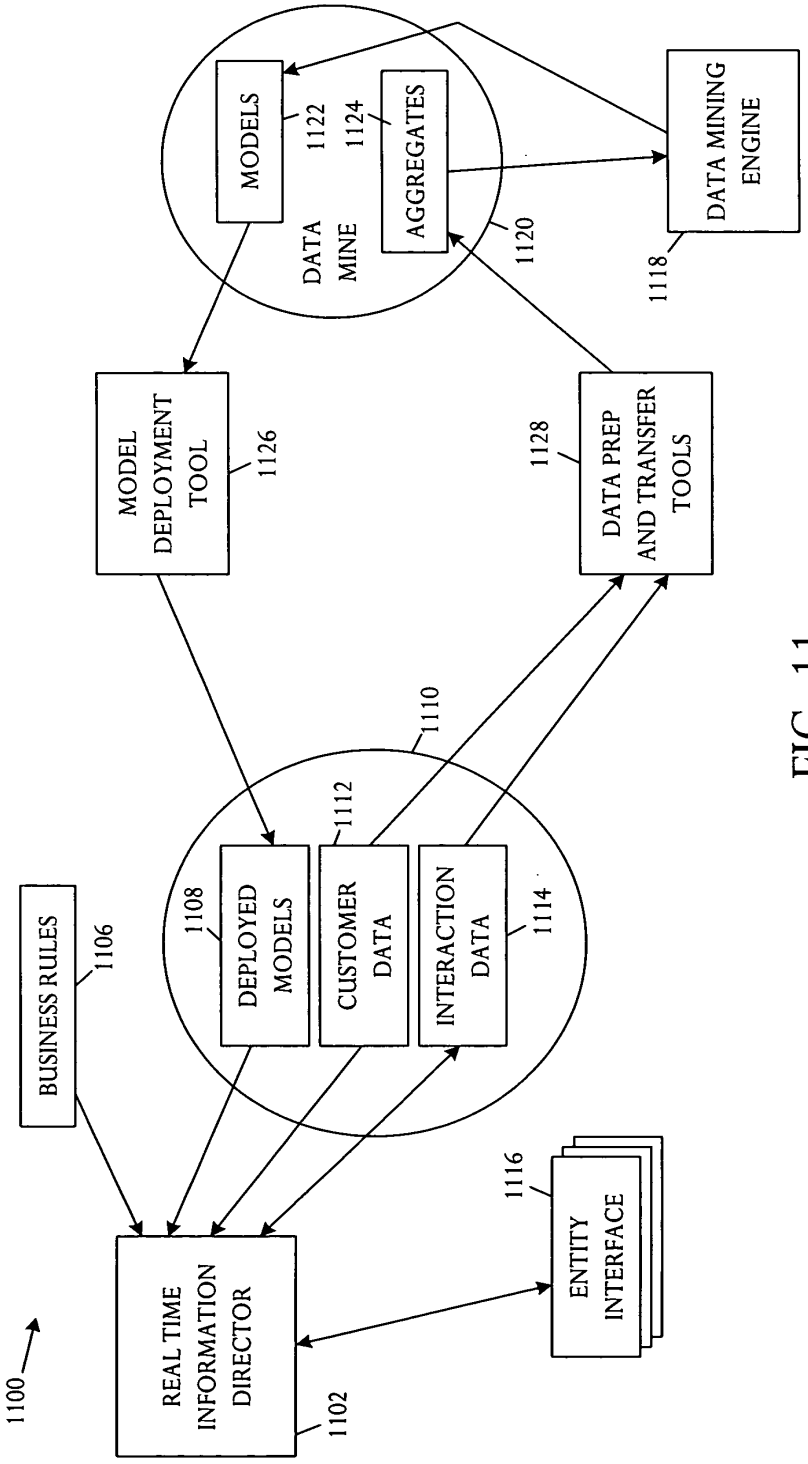

1100
REAL TIME INFORMATION DIRECTOR
1102
BUSINESS RULES
1106
ENTITY INTERFACE
1116
DEPLOYED MODELS
1108
CUSTOMER DATA
1112
INTERACTION DATA
1114
1110
MODEL DEPLOYMENT TOOL
1126
DATA PREP AND TRANSFER TOOLS
1128
DATA MINE
MODELS
1122
AGGREGATES
1124
1120
DATA MINING ENGINE
1118

FIG. 11

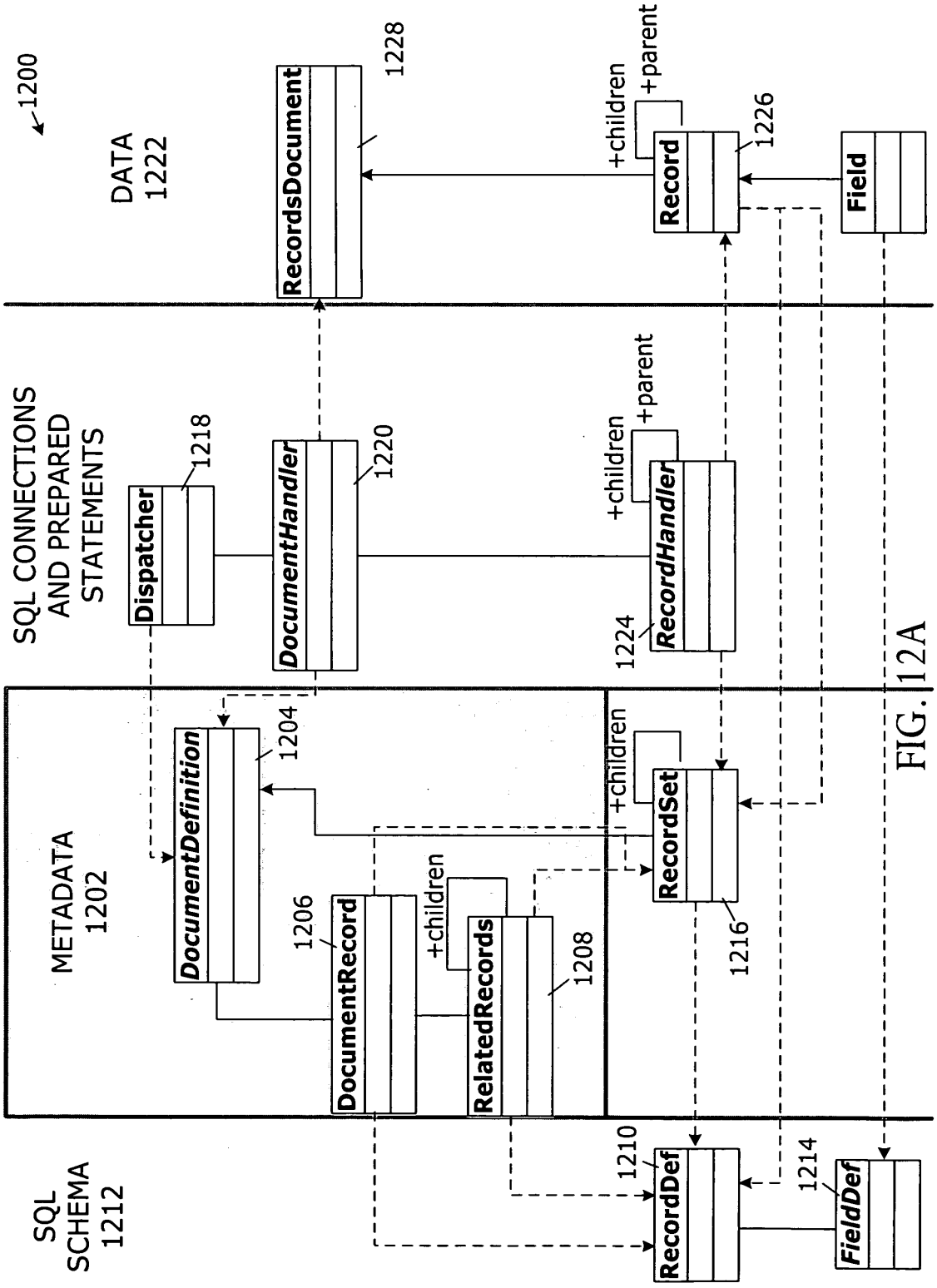

1200
DATA
1222
RecordsDocument
1228
+children
+parent
Record
1226
Field
SQL CONNECTIONS
AND PREPARED
STATEMENTS
Dispatcher
1218
DocumentHandler
1220
+children
+parent
RecordHandler
1224
METADATA
1202
DocumentDefinition
1204
1206
DocumentRecord
+children
RelatedRecords
1208
+children
RecordSet
1216
SQL
SCHEMA
1212
1210
RecordDef
1214
FieldDef

FIG. 12A

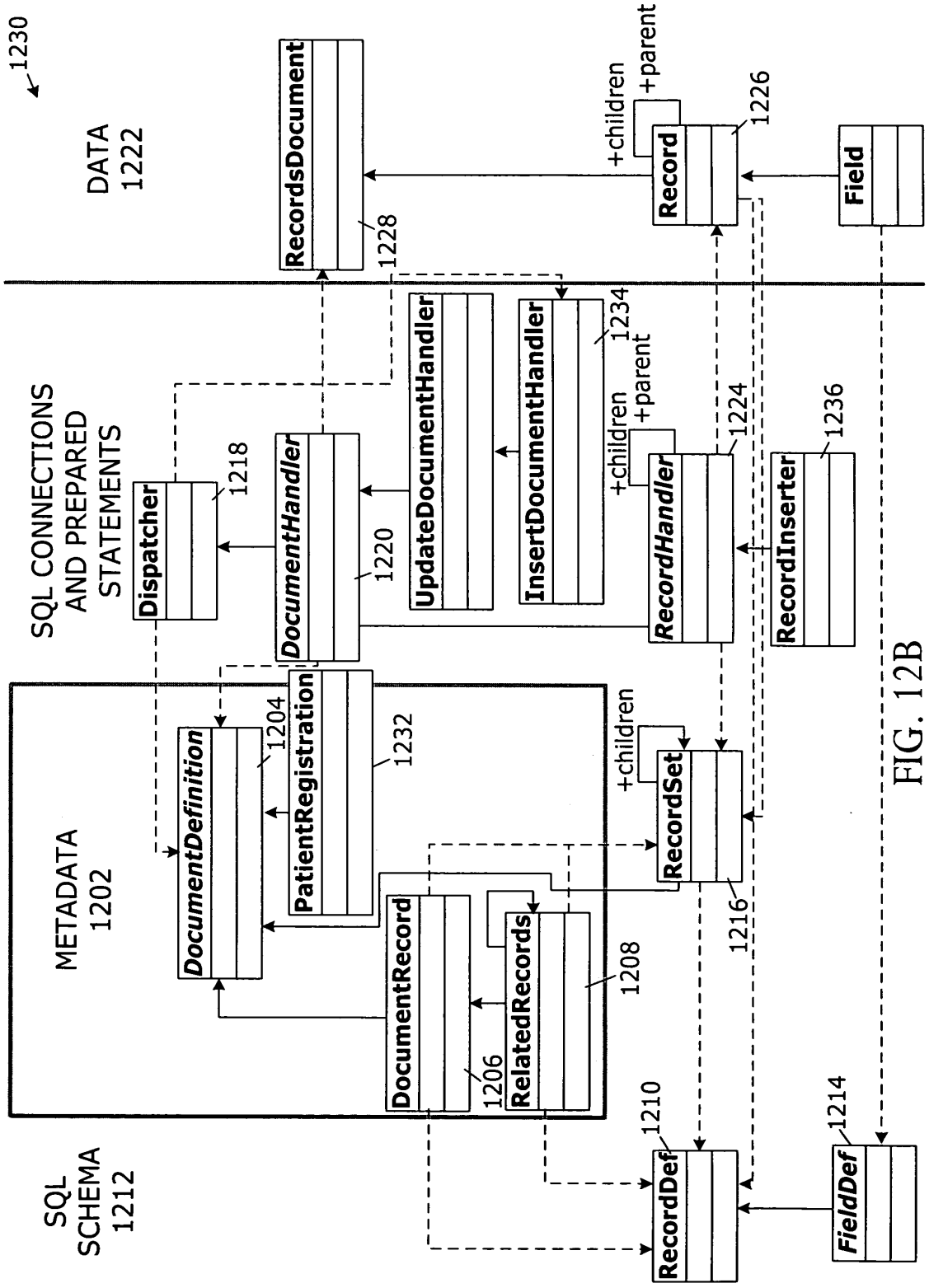

1230
DATA
1222
RecordsDocument
1228
+children
+parent
Record
1226
Field
SQL CONNECTIONS
AND PREPARED
STATEMENTS
Dispatcher
1218
DocumentHandler
1220
UpdateDocumentHandler
InsertDocumentHandler
1234
+children
+parent
RecordHandler
1224
RecordInserter
1236
METADATA
1202
DocumentDefinition
1204
PatientRegistration
1232
+children
RecordSet
1216
DocumentRecord
1206
RelatedRecords
1208
SQL
SCHEMA
1212
RecordDef
1210
FieldDef
1214

FIG. 12B

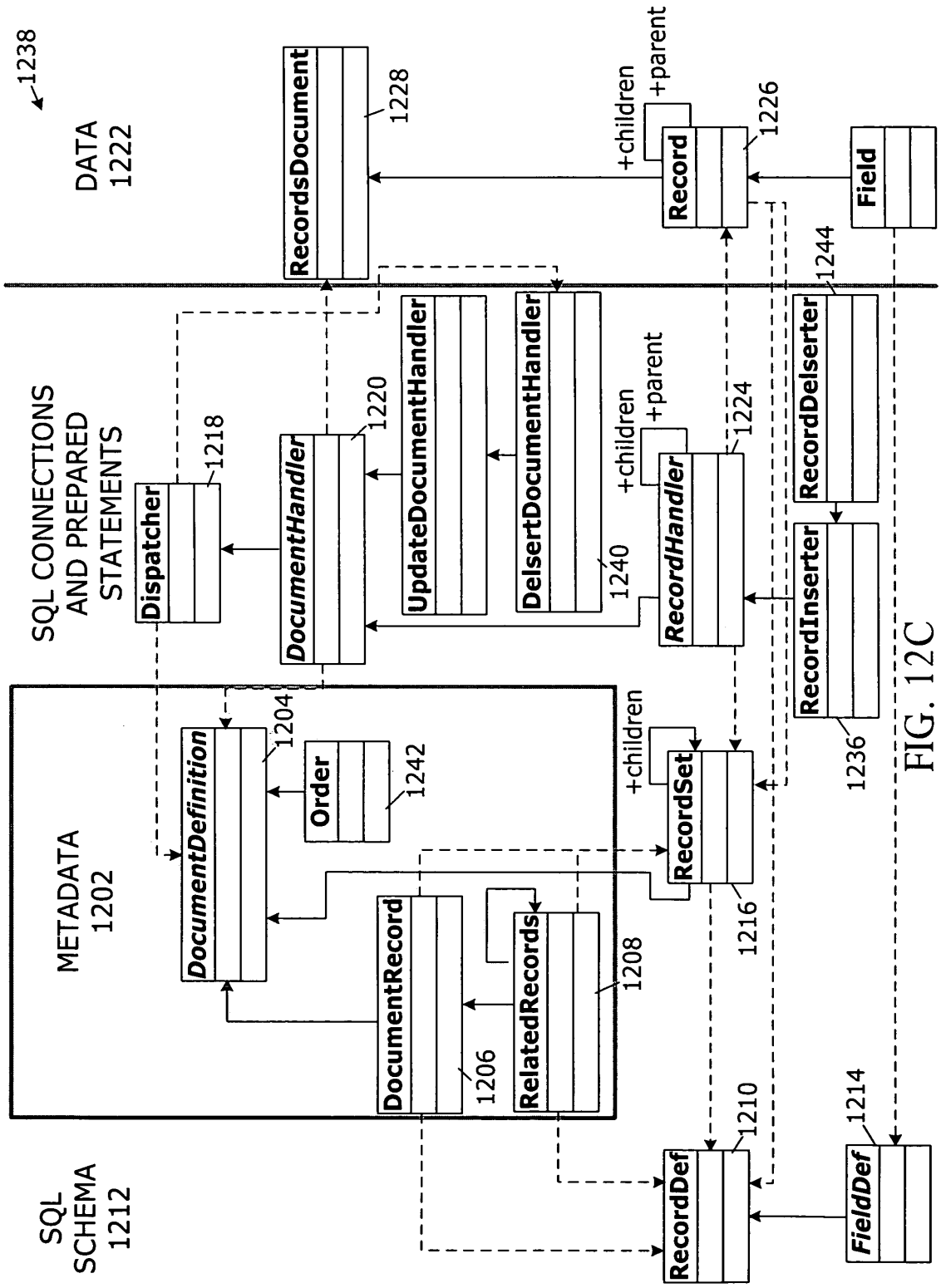

1238
DATA
1222
RecordsDocument
1228
+children
+parent
Record
1226
Field
1244
SQL CONNECTIONS
AND PREPARED
STATEMENTS
Dispatcher
1218
DocumentHandler
1220
UpdateDocumentHandler
DelsertDocumentHandler
1240
+children
+parent
RecordHandler
1224
RecordDelserter
RecordInserter
1236
METADATA
1202
DocumentDefinition
1204
Order
1242
+children
RecordSet
1216
DocumentRecord
1206
RelatedRecords
1208
SQL
SCHEMA
1212
RecordDef
1210
FieldDef
1214

FIG. 12C

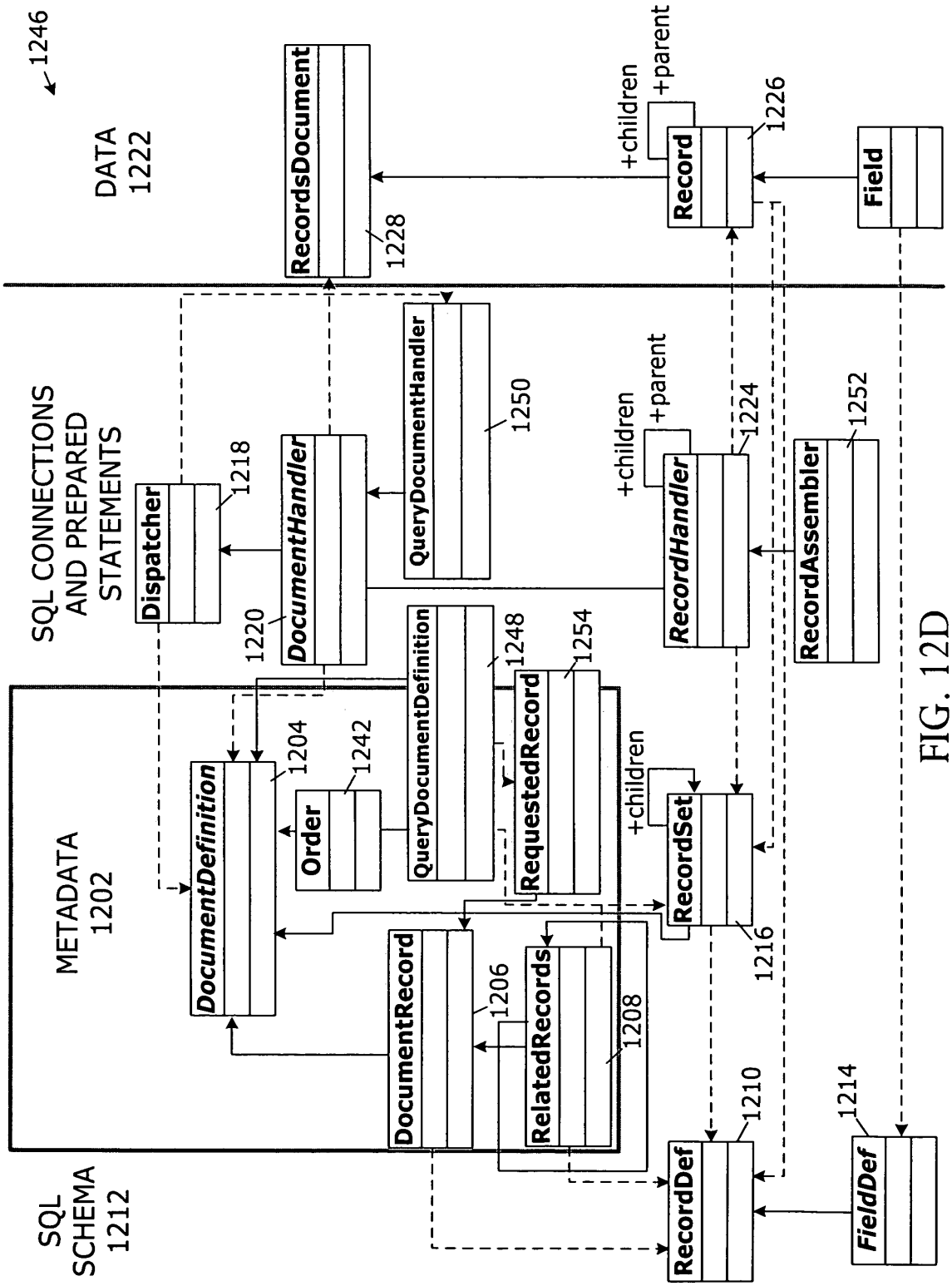

1246
DATA
1222
RecordsDocument
1228
+children
+parent
Record
1226
Field
SQL CONNECTIONS
AND PREPARED
STATEMENTS
Dispatcher
1218
DocumentHandler
1220
QueryDocumentHandler
1250
+children
+parent
RecordHandler
1224
RecordAssembler
1252
METADATA
1202
DocumentDefinition
1204
Order
1242
QueryDocumentDefinition
1248
RequestedRecord
1254
+children
RecordSet
1216
DocumentRecord
1206
RelatedRecords
1208
SQL
SCHEMA
1212
RecordDef
1210
FieldDef
1214

FIG. 12D

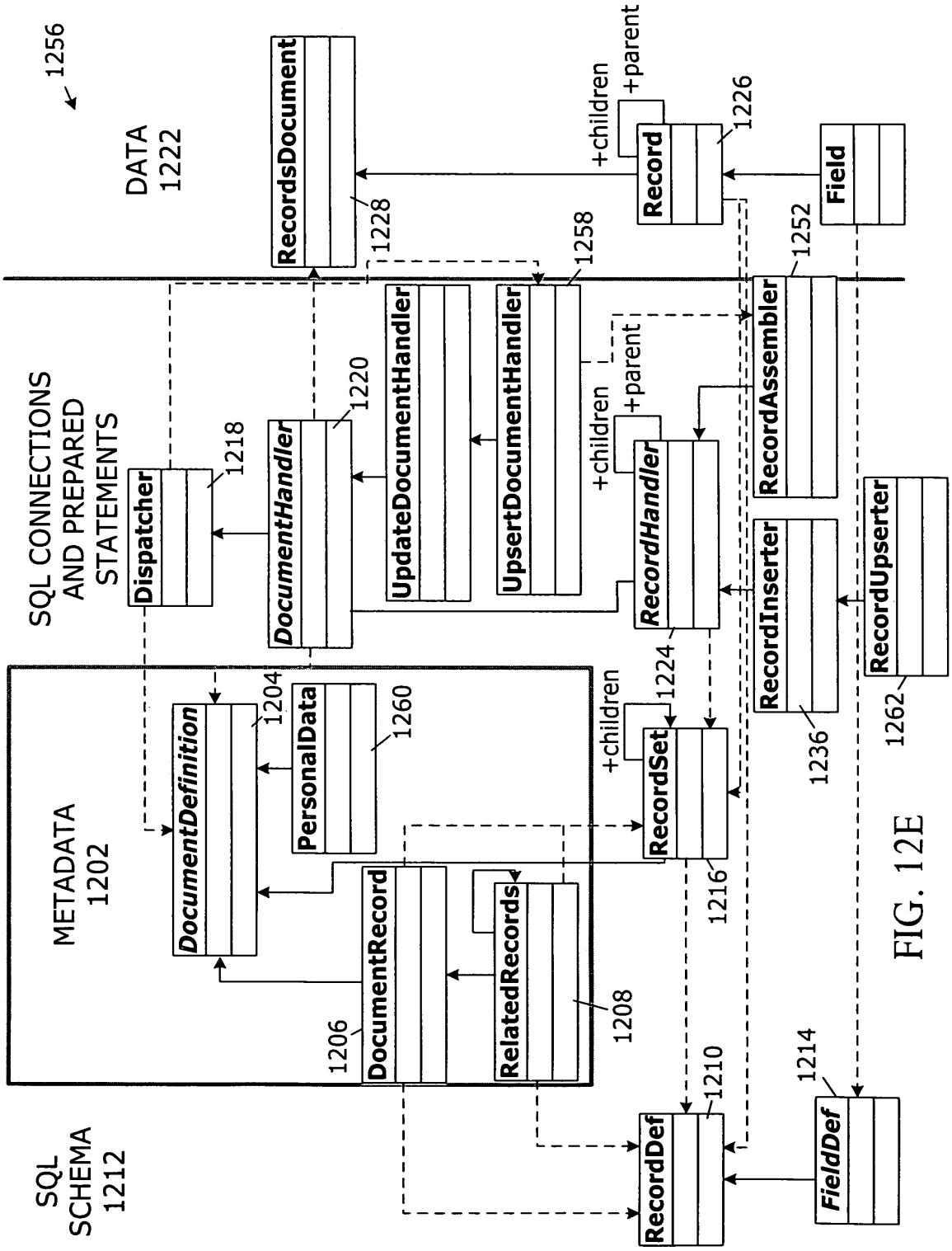

1256
DATA
1222
RecordsDocument
1228
+children
+parent
Record
1226
Field
1252
1258
SQL CONNECTIONS
AND PREPARED
STATEMENTS
Dispatcher
1218
DocumentHandler
1220
UpdateDocumentHandler
UpsertDocumentHandler
+children
+parent
RecordHandler
RecordAssembler
RecordInserter
RecordUpserter
1236
1262
1224
METADATA
1202
DocumentDefinition
1204
PersonalData
1260
+children
RecordSet
1216
DocumentRecord
1206
RelatedRecords
1208
SQL
SCHEMA
1212
RecordDef
1210
FieldDef
1214

FIG. 12E

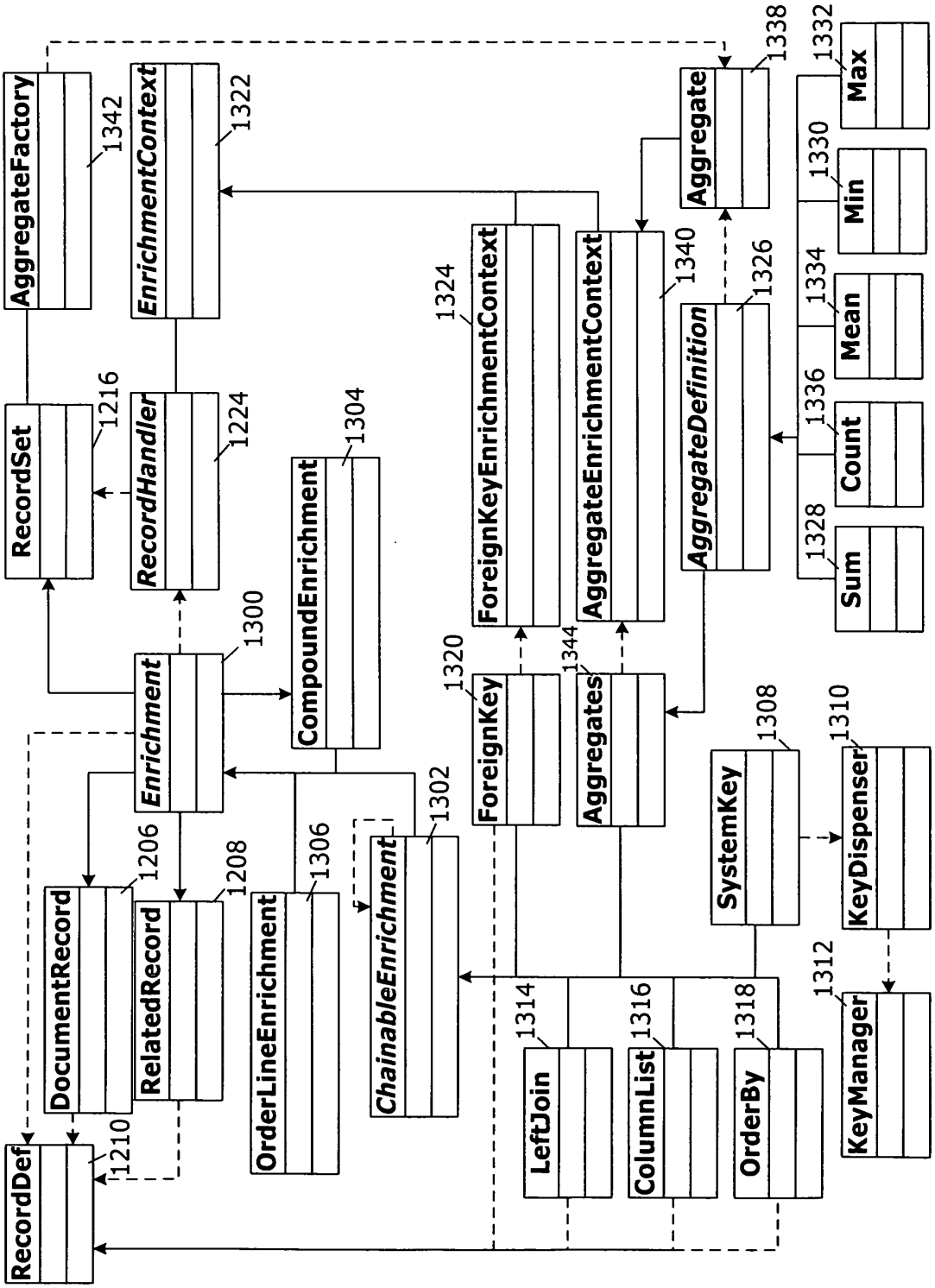

AggregateFactory
1342
EnrichmentContext
1322
Aggregate
1338
Max
1332
Min
1330
Mean
1334
Count
1336
Sum
1328
AggregateDefinition
1326
AggregateEnrichmentContext
1340
ForeignKeyEnrichmentContext
1324
RecordSet
1216
RecordHandler
1224
CompoundEnrichment
1304
Enrichment
1300
ForeignKey
1320
Aggregates
1344
SystemKey
1308
KeyDispenser
1310
DocumentRecord
1206
RelatedRecord
1208
OrderLineEnrichment
1306
ChainableEnrichment
1302
LeftJoin
1314
ColumnList
1316
OrderBy
1318
KeyManager
1312
RecordDef
1210

FIG. 13

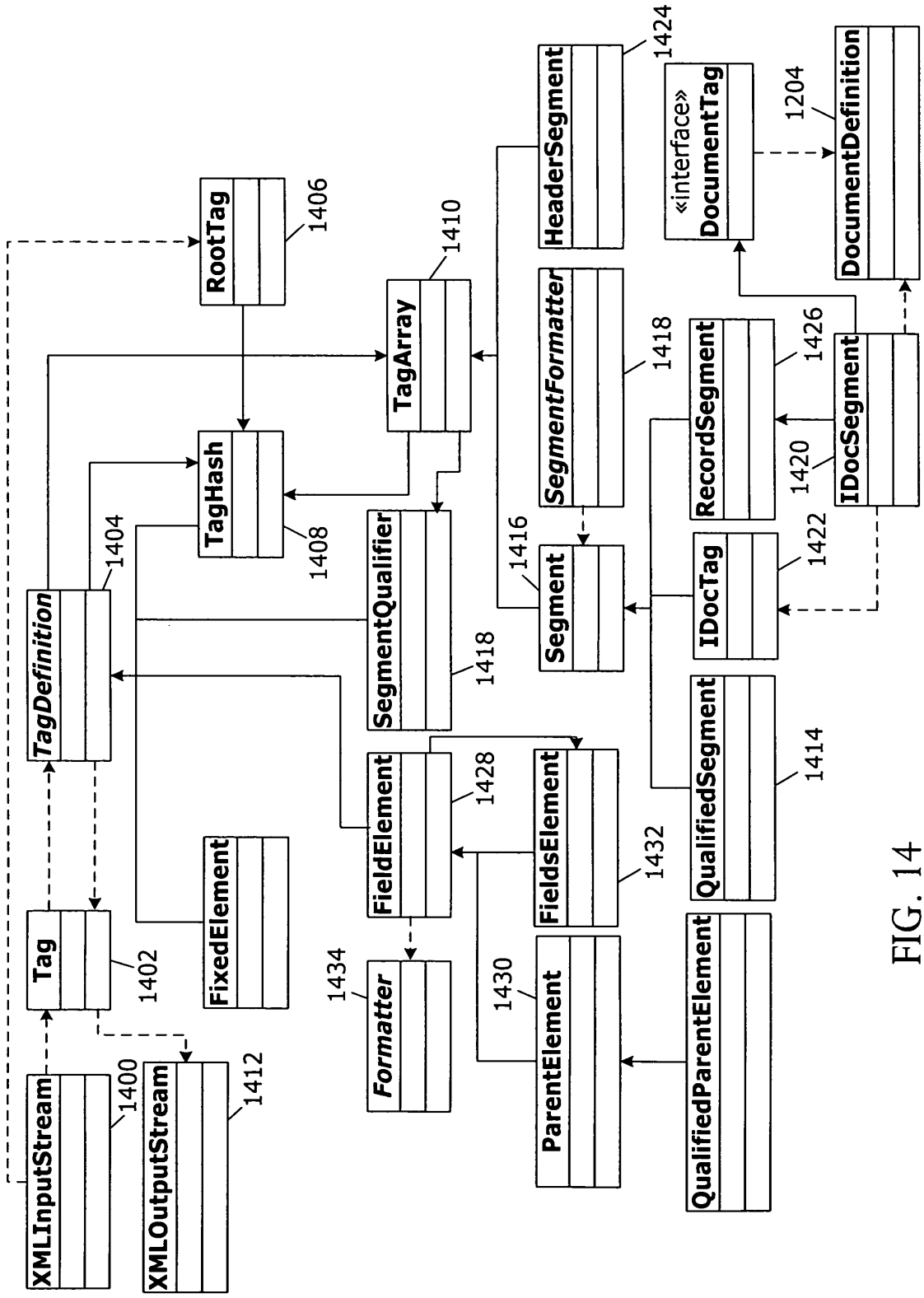

XMLInputStream
1400
XMLOutputStream
1412
Tag
1402
TagDefinition
1404
RootTag
1406
TagHash
1408
TagArray
1410
FixedElement
FieldElement
1428
SegmentQualifier
1418
Formatter
1434
ParentElement
1430
FieldsElement
1432
HeaderSegment
1424
SegmentFormatter
1418
Segment
1416
«interface»
DocumentTag
1204
DocumentDefinition
RecordSegment
1426
IDocSegment
1420
IDocTag
1422
QualifiedSegment
1414
QualifiedParentElement

FIG. 14

PATENT APPLICATION

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

ATTORNEY DOCKET NO. 200400406-1

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:

DATA SERVICES HANDLER

the specification of which is attached hereto unless the following box is checked:

( ) was filed on _____ as US Application No. or PCT International Application Number _____ 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(s) referred to above. I acknowledge the duty to disclose all information which is material to patentability as defined in 37 CFR 1.56.

Foreign Application(s) and/or Claim of Foreign Priority

I hereby claim foreign priority benefits under Title 35, United States Code Section 119 of any foreign application(s) for patent or inventor(s) certificate listed below and have also identified below any foreign application for patent or inventor(s) certificate having a filing date before that of the application on which priority is claimed:

COUNTRY	APPLICATION NUMBER	DATE FILED	PRIORITY CLAIMED UNDER 35 U.S.C. 119
			YES: ___ NO: ___
			YES: ___ NO: ___

Provisional Application

I hereby claim the benefit under Title 35, United States Code Section 119(e) of any United States provisional application(s) listed below:

APPLICATION NUMBER	FILING DATE

U. S. Priority Claim

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, Section 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

APPLICATION NUMBER	FILING DATE	STATUS (patented/pending/abandoned)

POWER OF ATTORNEY:

As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith:

Send Correspondence to:

Customer Number 022879

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

Direct Telephone Calls to:

Gerald E. Laws

(281) 518-7159

Even though the attorney is listed, please associate HP's Customer Number 022879 with this case.

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 issued thereon.

Full Name of Inventor: Steven R. Carr **Citizenship:** USA

Residence: San Jose, CA

Post Office Address: 3115 Terra Cotta Ct., San Jose, CA 95135

Steven R Carr 6/8/04

Inventor's Signature Date

Rev 05/04 (DecPwr) (Use Page Two For Additional Inventor(s) Signature(s)) Page 1 of 2

BEST AVAILABLE COPY

DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION (continued)

ATTORNEY DOCKET NO. 200400406-1

Full Name of # 2 joint inventor: Venkatakrishnan Muthuswamy Citizenship: India

Residence: Cupertino, CA

Post Office Address: 10870 N. Stelling Rd. #25A, Cupertino, CA 95014

M. Venkatakrishnan — 6/4/04

Inventor's Signature — Date

Full Name of # 3 joint inventor: Tudor Har Citizenship: USA

Residence: Santa Clara, CA

Post Office Address: 2541 South Park Lane, Santa Clara, CA 95051

Tudor Har — 6/8/04

Inventor's Signature — Date

Full Name of # 4 joint inventor: _____ Citizenship: _____

Residence: _____

Post Office Address: _____

Inventor's Signature — Date

Full Name of # 5 joint inventor: _____ Citizenship: _____

Residence: _____

Post Office Address: _____

Inventor's Signature — Date

Full Name of # 6 joint inventor: _____ Citizenship: _____

Residence: _____

Post Office Address: _____

Inventor's Signature — Date

Full Name of # 7 joint inventor: _____ Citizenship: _____

Residence: _____

Post Office Address: _____

Inventor's Signature — Date

Full Name of # 8 joint inventor: _____ Citizenship: _____

Residence: _____

Post Office Address: _____

Inventor's Signature — Date

Rev 05/04 (DecPwr)

(Use Page Two For Additional Inventor(s) Signature(s))

BEST AVAILABLE COPY

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 293 of 302

APPLICATION DATA SHEET

Application Information

Application number::	
Filing Date::	
Application Type::	Regular
Subject Matter::	Utility
Suggested Classification::	
Suggested Group Art Unit::	
CD-ROM or CD-R?::	None
Title::	Data Services Handler
Attorney Docket Number::	200400406-1
Request for Early Publication?::	No
Request for Non-Publication?::	No
Suggested Drawing Figure::	
Total Drawing Sheets::	24
Small Entity?::	No
Petition Included?::	No
Petition Type::	
Secrecy Order in Parent Appl.?::	No

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 294 of 302

Applicant Information

Applicant Authority Type::	Inventor
Primary Citizenship Country::	USA
Status::	Full Capacity
Given Name::	Steven
Middle Name::	R.
Family Name::	Carr
Name Suffix::	
City of Residence::	San Jose
State or Province of Residence::	CA
Country of Residence::	USA
Street of Mailing Address::	3115 Terra Cotta Ct.
City of Mailing Address::	San Jose
State or Province of Mailing Address::	CA
Country of Mailing Address::	USA
Postal or Zip Code of Mailing Address::	95135

Applicant Authority Type::	Inventor
Primary Citizenship Country::	INDIA
Status::	Full Capacity
Given Name::	Venkatakrishnan
Middle Name::	
Family Name::	Muthuswamy
Name Suffix::	
City of Residence::	Cupertino

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 295 of 302

State or Province of Residence::	CA
Country of Residence::	USA
Street of Mailing Address::	10870 N. Stelling Rd. #25A
City of Mailing Address::	Cupertino
State or Province of Mailing Address::	CA
Country of Mailing Address::	USA
Postal or Zip Code of Mailing Address::	95014

Applicant Authority Type::	Inventor
Primary Citizenship Country::	USA
Status::	Full Capacity
Given Name::	Tudor
Middle Name::	
Family Name::	Har
Name Suffix::	
City of Residence::	Santa Clara
State or Province of Residence::	CA
Country of Residence::	USA
Street of Mailing Address::	2541 South Park Lane
City of Mailing Address::	Santa Clara
State or Province of Mailing Address::	CA
Country of Mailing Address::	USA
Postal or Zip Code of Mailing Address::	95051

Initial June 14, 2004

Correspondence Information

Correspondence Customer Number::	022879
Phone Number::	(281) 518-7159
Fax Number::	(281) 514-8332
E-Mail Address::	gerald.laws@hp.com

Representative Information

Representative Customer Number::	022879

[This application has no priority claims]

Assignee Information

Assignee name::	Hewlett-Packard Development Company, L.P.
Street of Mailing Address::	20555 S.H 249
City of Mailing Address::	Houston
State or Province of Mailing Address::	TX
Country of Mailing Address::	USA
Postal or Zip Code of Mailing Address::	77070

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200400406-1

IN THE
UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s):	Steven R. Carr et al.	**Confirmation No.:**	
Application No.:		**Examiner:**	
Filing Date:	June 14, 2004	**Group Art Unit:**	
Title:	DATA SERVICES HANDLER		

Mail Stop Amendment
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

INFORMATION DISCLOSURE STATEMENT

Sir:

This Information Disclosure Statement is submitted:

(X) under 37 CFR 1.97(b), or
(Within three months of filing national application; or date of entry of national application; or before mailing date of first office action on the merits; whichever occurs last)

( ) under 37 CFR 1.97(c) together with either a:
( ) Statement under 37 CFR 1.97(e), or
( ) a $180.00 fee under 37 CFR 1.17(p), or
(After the CFR 1.97 (b) time period, but before final action or notice of allowance, whichever occurs first)

( ) under 37 CFR 1.97 (d) together with a:
( ) Statement under 37 CFR 1.97(e)(1) or (2), and
( ) a $180.00 fee set forth in 37 CFR 1.17(p).
(Filed after final action, a notice of allowance, on or before payment of the issue fee)

Please charge to Deposit Account **08-2025** the sum of $0.00 . At any time during the pendency of this application, please charge any fees required or credit any overpayment to Deposit Account **08-2025** pursuant to 37 CFR 1.25.

(X) Applicant(s) submit herewith Form PTO 1449 - Information Disclosure Statement together with any required copies of patents, publications or other information of which applicant(s) are aware, which applicant(s) believe(s) may be material to the examination of this application and for which there may be a duty to disclose in accordance with 37 CFR 1.56.

( ) A concise explanation of the relevance of foreign language patents, foreign language publications and other foreign language information listed on PTO Form 1449, as presently understood by the individuals(s) designated in 37 CFR 1.56 (c) most knowledgeable about the content is given on the attached sheet, or where a foreign language patent is cited in a search report or other action by a foreign patent office in a counterpart foreign application, an English language version of the search report or action which indicates the degree of relevance found by the foreign office is listed on form PTO 1449 and is enclosed herewith.

It is requested that the information disclosed herein be made of record in this application.

"Express Mail" label no. **EU 959884178 US**

Date of Deposit **June 14, 2004**

I hereby certify that this is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to: Commissioner for Patents, Alexandria, VA 22313-1450.

By _____

Typed Name: **Joy C. Ngo**

Rev 05/04 (IDSXML)

Respectfully submitted,

Steven R. Carr et al.

By _____

Ken J. Koestner

Attorney/Agent for Applicant(s)
Reg. No. **33,004**

Date: **June 14, 2004**

Telephone No.: **(949) 251-0250**

PATENT APPLICATION

Sheet 1 of 1

FORM PTO-1449 LIST OF PATENTS AND PUBLICATIONS FOR APPLICANT'S INFORMATION DISCLOSURE STATEMENT (Use several sheets if necessary)	ATTY. DOCKET NO. **200400406-1**	APPLICATION NO.	CONFIRMATION NO.
	APPLICANT **Steven R. Carr et al.**		
	FILING DATE **June 14, 2004**	GROUP	

REFERENCE DESIGNATION **U.S. PATENT DOCUMENTS**

EXAMINER INITIAL		DOCUMENT NUMBER	PUBLICATION DATE	NAME	Pages, Columns, Lines Where Relevant Passages or Figures Appear
	1A	6,157,707	12-05-2000	Baulier et al.	
	1B	6,163,604	Dec. 19, 2000	Baulier et al.	
	1C	6,233,575 B1	May 15, 2001	Agrawal et al.	
	1D	6,240,411 B1	May 29, 2001	Thearling	
	1E	6,317,700 B1	Nov. 13, 2001	Bagne	
	1F	6,480,844 B1	Nov. 12, 2002	Cortes et al.	
	1G	6,484,163 B1	Nov. 19, 2002	Lawrence et al.	
	1H	6,519,576 B1	Feb. 11, 2003	Freeman	
	1I	6,557,008 B1	Apr. 29, 2003	Temple, III et al.	
	1J	6,571,279 B1	May 27, 2003	Herz et al.	
	1K				

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	PUBLICATION DATE	NAME OF PATENTEE OR APPLICANT	Pages/Columns/Lines Where Relevant Passages/Figures Appear	Check if Translation attached
	1L					
	1M					
	1N					
	1O					
	1P					

OTHER REFERENCES (including Author, Title, Date, Pertinent Pages, etc.)

	1Q	
	1R	
	1S	

EXAMINER	DATE CONSIDERED

Rev 05/04 (PTO1449)

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2003

Application or Docket Number: 10868434

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	37	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	37 minus 20=	* 17
INDEPENDENT CLAIMS	3 minus 3 =	* 5
MULTIPLE DEPENDENT CLAIM PRESENT		☐

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

SMALL ENTITY TYPE ☐		OR	OTHER THAN SMALL ENTITY	
RATE	FEE		RATE	FEE
BASIC FEE	385.00	OR	BASIC FEE	770.00
X$ 9=		OR	X$18=	342
X43=		OR	X86=	430
+145=		OR	+290=	
TOTAL		OR	TOTAL	1542

CLAIMS AS AMENDED - PART II

AMENDMENT A

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

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

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

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

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

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
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.

SAP - Exhibit 1102
SAP v. Valtrus et al. - IPR2025-00416
Page 300 of 302

PATENT APPLICATION SERIAL NO. _____

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

06/21/2004 HMARZI1 00000033 082025 10868434

01 FC:1001 770.00 DA
02 FC:1202 342.00 DA
03 FC:1201 430.00 DA

PTO-1556
(5/87)

*U.S. Government Printing Office: 2001 — 481-697/59173

23026 U.S.PTO
061404

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, Colorado 80527-2400

PATENT APPLICATION

ATTORNEY DOCKET NO. 200400406-1

IN THE U.S. PATENT AND TRADEMARK OFFICE
Patent Application Transmittal Letter

22141 U.S.PTO
10/868434
061404

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

Transmitted herewith for filing under 37 CFR 1.53(b) is a(n): (X) Utility ( ) Design
(X) original patent application,
( ) continuation-in-part application

INVENTOR(S): **Steven R. Carr et al.**

TITLE: **DATA SERVICES HANDLER**

Enclosed are:

(X) The Declaration and Power of Attorney. (X) signed ( ) unsigned or partially signed

(X) 24 sheets of drawings (one set) ( ) Associate Power of Attorney

( ) Form PTO-1449 (X) Information Disclosure Statement and Form PTO-1449

( ) Priority document(s) (X) (Other) - Application Data Sheet (fee $_____)
- Recordation Form Cover Sheet
- Assignment

CLAIMS AS FILED BY OTHER THAN A SMALL ENTITY				
(1) FOR	(2) NUMBER FILED	(3) NUMBER EXTRA	(4) RATE	(5) TOTALS
TOTAL CLAIMS	39 — 20	19	X $18	$ 342
INDEPENDENT CLAIMS	8 — 3	5	X $86	$ 430
ANY MULTIPLE DEPENDENT CLAIMS	0		$290	$ 0
		BASIC FEE: Design ($340.00); Utility ($770.00)		$ 770
			TOTAL FILING FEE	$ 1,542
			OTHER FEES	$
			TOTAL CHARGES TO DEPOSIT ACCOUNT	$ 1,542

Charge $ 1,542 to Deposit Account 08-2025. At any time during the pendency of this application, please charge any fees required or credit any over payment to Deposit Account 08-2025 pursuant to 37 CFR 1.25. Additionally please charge any fees to Deposit Account 08-2025 under 37 CFR 1.16 through 1.21 inclusive, and any other sections in Title 37 of the Code of Federal Regulations that may regulate fees. A duplicate copy of this sheet is enclosed.

"Express Mail" label no. EU 959884178 US

Date of Deposit June 14, 2004

I hereby certify that this is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to: Commissioner for Patents, Alexandria, VA 22313-1450.

By _____

Typed Name: Joy C. Ngo

Respectfully submitted,

Steven R. Carr et al.

By _____

Ken J. Koestner

Attorney/Agent for Applicant(s)
Reg. No. 33,004

Date: June 14, 2004

Telephone No.: (949) 251-0250

Rev 05/04 (TransNew)

- Attach as First Page to Transmitted Papers -