

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
20 July 2006 (20.07.2006)

PCT

(10) International Publication Number
WO 2006/076520 A2

(51) International Patent Classification:
G06F 7/00 (2006.01)

(71) Applicant: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, New York 10504 (US).

(21) International Application Number:
PCT/US2006/001133

(72) Inventors: **ABRAMS, Carl Edward**; 425 Hardscrabble Road, Briarcliff Manor, New York 10510 (US). **ADINOLFI, Ronald Emmett**; 172 Wildflower Circle, Yardley, Pennsylvania 19067 (US). **CALUSINSKI JR., Edward Patrick**; 458 Briarcliff Lane, Bartlett, Illinois 60103 (US). **CROWLEY, Cornelius Edward**; 27 Blackwell Avenue, Morristown, New Jersey 07960 (US). **FLEMING JR., Michael Edward**; 11 Waterside Close, Eastchester, New York 10709 (US). **GLASSER, Teresa Anne**; 1 Central Park West, New York, New York 10023 (US). **GROMADA, Jennifer Susan**; 917 Lawrenceville Road, Princeton, New Jersey 08540 (US). **HRABROV, Max**; 4105 Bedford Ave, Brooklyn, New York 11229 (US). **HUNT, Guernsey Douglass Holloway**; 31 Wellington Court, Yorktown Heights, New York 10598 (US). **JONES, Kenneth Lee**; 136 Oak Blvd., Lansdale, Pennsylvania 19446 (US). **MEHTA, Sugandh**; 34 Medford Place, Wayne, New Jersey 07470 (US). **PARR,**

(22) International Filing Date: 13 January 2006 (13.01.2006)

(25) Filing Language: English

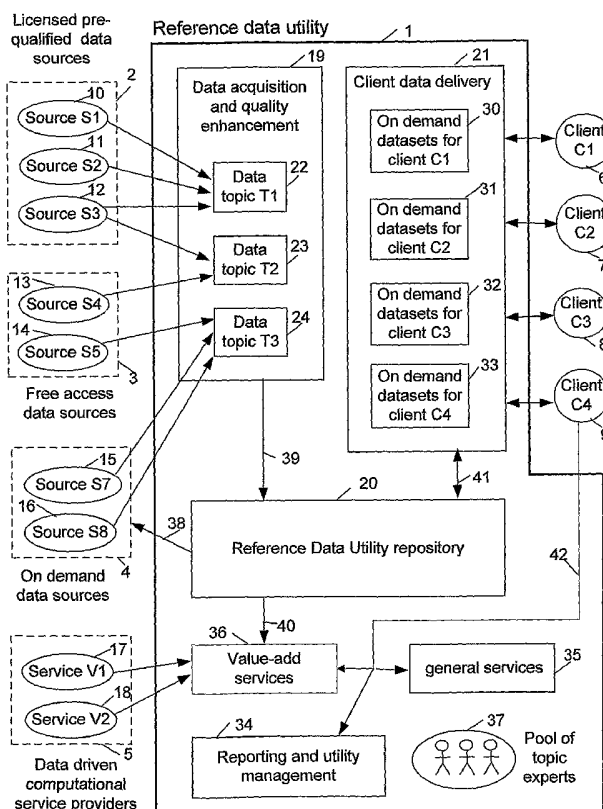
(26) Publication Language: English

(30) Priority Data:

60/644,045	14 January 2005 (14.01.2005)	US
60/648,497	31 January 2005 (31.01.2005)	US
60/654,376	18 February 2005 (18.02.2005)	US
60/694,815	28 June 2005 (28.06.2005)	US
11/318,355	22 December 2005 (22.12.2005)	US
11/318,425	22 December 2005 (22.12.2005)	US
11/318,426	22 December 2005 (22.12.2005)	US
11/318,428	22 December 2005 (22.12.2005)	US

[Continued on next page]

(54) Title: SHARABLE MULTI-TENANT REFERENCE DATA UTILITY AND REPOSITORY, INCLUDING VALUE ENHANCEMENT AND ON-DEMAND DATA DELIVERY AND METHODS OF OPERATION



(57) Abstract: A multi-source multi-tenant reference data utility (1) and methods for forming and maintaining the same, delivering high quality reference data in response to requests from clients (6, 7, 8, 9), implemented using a shared infrastructure, and also providing added value services using the client's reference data. Included are data cleansing and quality assurance of the received data with full tracking of the sourcing of each value, storage of resulting entity values in a repository (20) which allows retrievals and enforces source based entitlements of clients, and delivery of retrieved data in the form of on demand datasets (30, 31, 32, 33) supporting a wide range of client application needs. An advantageous implementation has additional services for reporting on data quality and usage, a selection of value adding data driven computations and business document storage. The shared infrastructure and amortizing the costs of data quality assurance, assures that better quality data at lower cost is delivered.



Francis Nicholas; 250 West 94th Street, New York, New York 10025 (US). **ORANI, Aviv**; 292 N. Fifth Avenue, Highland Park, New Jersey 08904 (US). **RICE, Matthew Adam**; 1 Lost Hill Drive, Medway, Massachusetts 02053 (US).

(74) **Agent: AKER, David**; 23 Southern Road, Hartsdale, New York 10530 (US).

(81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG,

SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

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

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

1 SHARABLE MULTI-TENANT REFERENCE DATA UTILITY AND REPOSITORY,
2 INCLUDING VALUE ENHANCEMENT AND ON-DEMAND DATA DELIVERY AND
3 METHODS OF OPERATION
4
5

6 PRIORITY

7 This application claims priority, under 35 U.S.C. §119(e), from United States provisional
8 application serial numbers 60/644,045 filed on January 14, 2005; 60/648,497 filed on January
9 31, 2005; 60/654,376 filed on February 18, 2005; and 60/694,815 filed on June 28, 2005.

10 This application also claims priority from United States patent applications 11/318,355;
11 11/318,425; 11/318,426; and 11/318,428, all filed on December 22, 2005.

12 These applications are incorporated herein by reference in their entirety, for all purposes.
13

14 FIELD OF INVENTION

15 This invention is directed to the field of data management utility services. It is more
16 particularly directed to enabling on demand receipt, cleansing, enhancement, storage,
17 tracking and provision of business data in the context of a multi-source multi-tenant data
18 utility. The invention is also it is directed to a data repository which may be used in such
19 context, and other contexts. It is also directed to flexible, scalable delivery of on demand data
20 sets.
21

22 BACKGROUND

23 Financial markets reference data includes the descriptive information about financial
24 instruments, market evaluations, interested parties, and the corporate actions that impact
25 financial instruments. Reference data forms the shared basis for financial transaction
26 processing, decision making, risk measurement, instrument and portfolio pricing, and the
27 functioning of financial markets trading operations. Included are thousands of data items,
28 ranging from name and address information and tax identification to contingent claim
29 schedules, transfer agent details, depository eligibility and tax treaty implications. One of the
30 problems the industry faces is the absence of standards in naming, extending to how the
31 different types of reference data are described. Financial instrument data comprises the items
32 that describe what the instrument is, when, how and where it is traded, what is needed to
33 settle and clear transactions in the instrument, and the various regulatory and client reporting
34 requirements. Included in the alternate labels for financial instrument data are securities

instrument data, product data, and indicative data (indicative is also use by some as a term to refer to indicative pricing data). Party data describes entities involved in financial transactions, e.g. corporations, counterparties, clients, trading partners and individual investors. Included in the alternate labels for party data is business data, legal entity hierarchy data, client data, and counter party data. Corporate actions data reflects changes that are made to the legal structure or financial instruments of a corporation, such as ownership changes or stock splits. Here again alternate include corporate events and mandated events.

Financial market reference data may define characteristics of public entities, such as stock quotes, financial instrument definitions, corporate address and press releases, or of private entities including client identification, model-derived analytics and risk calculations.

Firms acquire reference data either by delivery via an exchange or data services vendor or by derivation through the application of calculations or models. Firms needing this data typically contract with a number of data vendors and pay licensing fees for access to the vendor's product. In addition to the capture and provision of raw data, many firms, including financial services firms, specialize in the creation of analytic data that is in turn propagated through the industry.

Financial markets reference data is horizontally embedded throughout the lifecycle of business processes conducted by financial firms and, as such, timely, accurate, high quality reference data has great value to these firms. Without it, a firm would be unable to process even the simplest of transactions for their clients or their internal financial management processes.

As an example, for a trade to be executed completely and accurately between financial organizations, all parties to the trade must have equivalent views of relevant reference data. A stock trade requires agreement on: (1) the definition and description of the instrument being traded; (2) the details of the trade and formal documentation of the transaction; and (3) counterparties participating in the process and delivery instructions. Organizations with incompatible reference data will require additional time and resources to resolve differences on each affected trade execution. The need for agreement on reference data is heightened in automated trading environments and during high trading volume periods.

1 Consequently, each financial firm requires ready access to a high quality reference database,
2 where base reference data may be augmented with the results of higher level analytic and
3 pricing computations and additional information, such as contact details and account
4 information. This information must be in a format that is easily and fully integrated across
5 their portfolio of business applications. Historically, firms have each built and maintained
6 their own stores of information or data in isolation from other firms. As firms grow, whether
7 organically or through acquisition, additional data silos are established or acquired. These
8 databases are typically maintained through a combination of automated data feeds from
9 external vendors, internal applications, and manual entries and adjustments.

10
11 Advances in technology and the availability of vendor data sources have significantly
12 increased the amount of information available to firms. As a result, firms have to sift through
13 large amounts of information that might differ depending on the source and timing of the
14 updates.

15
16 The fragmented ingestion and maintenance of financial markets reference data, decentralized
17 approaches to data management, multiple or redundant quality assurance activities, and
18 duplicative data stores have led to increased costs and operational inefficiency in the
19 acquisition and maintenance of reference data. Thus, at the corporate level, the data
20 management challenge is one of cost and quality arising from the overwhelming quantity of
21 data. Redundant purchases and validation, different formats/tools, inconsistent
22 formats/standards/data, and difficulties in changing and/or managing vendors all contribute to
23 inefficiencies.

24
25 This could cause decisions to be made on inaccurate information or differences in data used
26 by trading counterparties. These impacts are clearly exemplified in the findings of the Tower
27 Group resulting from their 2002 study of reference data in financial markets. For example, in
28 the area of trades processing, where on average, 16.4% of trades are rejected from automated
29 processing routines, Tower Group found that 45% of the exceptions (e.g. trades rejected from
30 automated processing routines) are due to faulty (incomplete, nonstandard, or inaccurate)
31 reference data ("TowerGroup Survey: Is the Securities Industry Making Progress on
32 Reference Data Management?" September 2002). In fact, failed trades resulting from
33 inaccurate reconciliation cost the domestic securities industry in excess of \$100 million per

1 year (IBM Institute for Business Value analysis). Although reference data comprise a
2 minority of the data elements in trade record, problems with the accuracy of this data
3 contribute to a disproportionate number of exceptions, clearly degrading straight through
4 processing (STP) rates.

5
6 Data inconsistency encountered by financial firms is discernable as erroneous or inconsistent
7 information. In many cases, data provided by external vendors contains errors, a fact which a
8 company may uncover by comparing data from multiple vendors or which may be revealed
9 as the result of using this data in an internal business process or in a transaction with an
10 external entity. Each data vendor has proprietary ways of representing data, due largely to a
11 lack of industry standards governing the representation of data. As well, financial services
12 firms utilize a variety of formats, including vendor or exchange-specific and proprietary
13 definitions, to define data within the enterprise.

14
15 While various data standardization initiatives are underway across the industry to agree on
16 standards for some data, none of the initiatives are mature. Although financial services firms
17 could realize significant improvements in transaction processing efficiencies from the
18 implementation of clear data standards, both vendors and securities firms have historically
19 viewed the anticipated retrofitting or adapting of existing applications to accept new data
20 formats as an impediment to widespread adoption.

21
22 Due to the overwhelming quantity and uneven quality of financial market data, financial
23 firms are obligated to commit significant attention and resources to the management of data
24 that, in many cases, provides them with no discernable competitive advantage.

25
26 In addition, recent regulatory changes require firms to store and track financial information
27 more diligently. For example, the Sarbanes-Oxley Act specifies strict requirements on the
28 transfer of information between financial services businesses, even within the departments of
29 a single firm.

30
31 As an industry, inconsistent levels of quality and lack of standards for financial markets
32 reference data reduce the efficiency and accuracy of communications between firms,
33 resulting in increased costs and higher levels of risk for all transaction participants. When

1 compounded by the multiple number of parties involved in the end-to-end execution of a
2 financial transaction, it is apparent that issues of data quality and standardization have
3 tremendous detrimental impact on the ability of the financial services industry to accomplish
4 straight through processing to a significant degree. The effect of this complexity is
5 exacerbated by the increasingly international scope of the business, as issues of cross-border
6 sovereignty; regulation and currency introduce incremental data elements as well as
7 additional variations of existing data.

8
9 All of these factors are providing additional impetus for financial firms to seek automated
10 assistance in gathering high quality data, tracking origin and data modification history, as
11 well as storing and managing access to that data and any additional information that may
12 have been created using the data.

13
14 Within financial services there are many current practices employed in organizing and
15 maintaining high quality reference data. Historically, firms have each built and maintained
16 their own stores of information or data in isolation from other firms. Financial instrument
17 descriptions and associated data are generally stored in databases referred to as the Product or
18 Security Master File. Party and customer data are generally stored in databases referred to as
19 the Customer Master File. A majority of Security and Customer master files are similar in
20 nature and content across firms.

21
22 Many financial service firms currently have decentralized, often incompatible, and
23 fragmented data stores. As firms grow, whether organically or through acquisition,
24 additional data silos are established or acquired. These data silos are populated by a variety of
25 data from multiple vendors through efforts that are rarely coordinated. A lack of enterprise-
26 wide integration prevents many business functions from fully realizing the value of much in-
27 house data. Further, this decentralized approach to data management frequently produces
28 redundant stores of identical data that are often created and updated by duplicate data feeds
29 paid for by separate organizations within a firm.

30
31 As a result of attempts to address such data management problems, some support for data
32 management outsourcing is available in the marketplace as a service to individual clients.

1 Some specific reference data management components, including repositories, are available
2 as well. However the current state-of-the-art of these offerings is:

- 3 - applicable only to a particular subset of reference data;
- 4 - not developed with multi-tenancy / multi-client support in mind;
- 5 - delivered as a one-off service to a single client; or
- 6 - implemented and priced as a stand-alone service for a single client.

7
8 Yet, a large portion of the work performed by, or on behalf of the above mentioned
9 organizations to manage their reference data, is in fact rather generic. As such, a lot of effort
10 associated with reference data management is duplicated across the financial industry sector,
11 as well as other industries. There remains therefore a need to establish a multi-tenant
12 reference data utility which could provide best practice data management and processing and
13 reduce costs to individual organizations through economies of scale. However, the
14 technology to build such a utility while properly dealing with certain complexities inherent in
15 the centralized utility approach (such as multi-source multi-tenant entitlement management)
16 is not currently available in the marketplace, and only single-client, localized approaches
17 exist.

18
19 Specific examples of localized technologies applicable include:

- 20 - standardization of base reference data model within one organization for use by its internal
21 departments;
- 22 - models and standardized formats for particular areas of financial reference data; and
- 23 - tools and automation to assist the entry of data into a data model for use by a single
24 organization.

25
26 There are a number of companies with existing technology and services offerings in the
27 financial services reference data management area which use this localized approach. The
28 solutions that these companies offer are generally targeted at solving the reference data
29 management problem of a single enterprise or a department within an enterprise, usually
30 within the domain of a narrowly defined problem. The software and services they provide
31 are normally installed, configured, customized and operated for a single client/department.
32 As a result, each customer implementation is effectively a dedicated, custom product
33 installation. As such, these offerings may be considered individual solutions to internal

1 reference data management problems and cannot provide economies of scale at the same
2 level that a multi-tenant capable solution can. Further, these solutions do not provide the
3 additional benefits afforded by a shared utility environment, such as turn-key data vendor
4 switching, on-demand billing, leveraged human capital, etc.

5
6 Isolated attempts have been made to use single client solutions to support multi-client
7 installations. However, in prior art, leveraging these solutions for multiple clients has
8 essentially required multiple duplication of single-client operations. These attempts have
9 generally not been successful within the financial services industry.

12 SUMMARY OF THE INVENTION

14 A. Reference Data Utility

15 The invention is a method, apparatus and software for forming and maintaining a multi-
16 source multi-tenant reference data utility delivering high quality reference data in response to
17 requests from clients, implemented using a shared infrastructure, and also providing added
18 value services using the client's reference data. The method includes data cleansing and
19 quality assurance of the received data with full tracking of the sourcing of each value, storage
20 of resulting entity values in a repository which allows retrievals and enforces source based
21 entitlements, and delivery of retrieved data in the form of on demand datasets supporting a
22 wide range of client application needs. An advantageous implementation has additional
23 services for reporting on data quality and usage, a selection of value adding data driven
24 computations and business document storage. By using a shared infrastructure and
25 amortizing the costs of data quality assurance across a plurality of clients, while ensuring that
26 clients only receive values from data sources to which they are licensed, this reference data
27 utility delivers better quality data at lower cost than other methods currently available.

28
29 Thus, a first aspect of the invention is directed to a reference data utility for serving a plurality of
30 recipients, comprising: data inputs for receiving unprocessed reference data from a plurality
31 of sources; a processor for processing the unprocessed reference data received so as to
32 generate processed reference data having an increased value; a repository for storing the
33 unprocessed reference data and the processed reference data; and an output generator for

1 generating output data for delivery to recipients, in accordance with specifications of
2 recipients; so that delivered output data contains at least one of unprocessed reference data
3 and processed reference data, that the recipient is entitled to receive; wherein the reference
4 data utility is scalable so as to support an increasing number of sources and an increasing
5 number of recipients. The reference data utility can be configured as a multi-tenant utility.
6 The reference data utility can be implemented as a system of shared resources. The shared
7 resources comprise at least one of the following: repositories, experts, processing,
8 communications links, and data storage facilities.

9
10 The reference data can further comprising means for tenants to perform self service
11 administration of their clients.

12
13 The repository can store a plurality of business documents, and the output generator can
14 provide as output a selected group of the documents. A data cleansing portion for cleansing
15 the unprocessed reference data can be provided. The reference data utility can further
16 comprise a memory portion for storing processed and unprocessed reference data and, with
17 each unprocessed or processed reference data element, a record of the data sources and
18 applied processing used to derive the element; said sourcing and processing determining the
19 entitlement of individual recipients to receive the element.

20
21 The recipients can be individuals granted entitlement to particular sources of reference data
22 and enhancement processes by at least one of a plurality of tenant organizations sharing use
23 of the reference data utility. The recipients are preferably selected from among different
24 business organizations and independent individuals who subscribe to selected portions of the
25 output data based on their entitlements.

26
27 The unprocessed reference data comprises information elements, and the reference data
28 utility further comprises means for annotating a plurality of the information elements with
29 sourcing information. The information elements have attributes, and the reference data utility
30 further comprises means for annotating the attributes with sourcing information. The
31 reference data utility may further comprise means for maintaining information about
32 entitlement of recipients to the information elements based on the sourcing information.

1 The reference data utility may be comprised of components located in geographically
2 dispersed regions. Preferably, the components located in one of the geographically dispersed
3 regions are sufficient to operate as an independent reference data utility. Each independent
4 reference data utility includes a local repository, and may further comprise communication
5 facilities for exchange of information between the local repositories. Each independent
6 reference data utility can be specialized to provide information pertaining to a particular
7 geographic region, and can use the communication facilities to obtain and provide
8 information from other independent reference data utilities in other geographic regions.

9
10 The reference data utility can further comprise an accuracy reporter for reporting accuracy
11 of processes performed by the reference data utility. It may also further comprise a
12 configuration manager for managing parameters of the reference data utility.

13
14 The configuration manager comprises at least one of: means for managing a number of
15 maximum allowable parallel data enhancement processes, means for managing types of
16 single-source cleansing processes applied during a data enhancement process, means for
17 managing types of cross-source processes applied during a data enhancement process, means
18 for managing rules to be applied during specific single-source cleansing processes, and
19 means for managing rules to be applied during specific cross-source processes.

20
21 The output generator can comprise: means for receiving at least one request from a recipient;
22 means for parsing the at least one request to extract a request specification; and means for
23 initiating at least one work flow to provide the output data to the recipient.

24
25 The invention is also directed to a method for operating a reference data utility for serving a
26 plurality of recipients, comprising: receiving unprocessed reference data inputs from a
27 plurality of sources; processing the unprocessed reference data received so as to generate
28 processed reference data having an increased value; storing the unprocessed reference data
29 and the processed reference data; and generating output data for specified recipients; so that
30 the output data contains only at least one of unprocessed reference data and processed
31 reference data, that the recipient is entitled to receive.

1 The method can further comprise configuring the reference data utility so as to be scalable
2 with respect to support for at least one of an increasing number of sources, an increasing
3 number of recipients, an increasing number of processes, and an increasing number and
4 complexity of entitlements. The method can further comprise storing a plurality of business
5 documents the repository, and generating as output a selected group of the documents.
6 Preferably, the method further comprises cleansing the unprocessed reference data. The
7 method further comprises storing access rights to sources, wherein the data that a recipient is
8 entitled to receive is defined by the access rights. The recipients are individuals granted
9 entitlement to particular sources of reference data and enhancement processes by at least one
10 of a plurality of tenant organizations sharing use of the reference data utility, the at least one
11 of the tenant organizations arranging independently with one or more data sources to have
12 entitlements to their data, and with the reference data utility, to have entitlement to the results
13 of applying specific data enhancement processes to other reference data, entitled to the at
14 least one tenant organization.

15
16 The unprocessed reference data comprises information elements, and the reference data
17 utility annotates a plurality of the information elements with sourcing information. The
18 information elements have attributes, and the reference data utility annotates the attributes
19 with sourcing information. The method further comprises maintaining information about
20 entitlement of recipients to the information elements, based on the sourcing information.

21
22 The method can further comprising utilizing apparatus located in geographically dispersed
23 regions. Components located in one of the geographically dispersed regions can be operated
24 as an independent reference data utility. Each independent reference data utility can include a
25 local repository, and the method can further comprise communicating information between
26 the local repositories. Each independent reference data utility can be specialized to provide
27 information pertaining to a particular geographic region, and the method can further comprise
28 communicating information from other independent reference data utilities in other
29 geographic regions.

30 The method may comprise reporting accuracy of processes performed by the reference data
31 utility. The accuracy of a source can be assessed by a combination of recording quality
32 enhancement actions on values received from a source; and comparing newly-arriving

1 reference values with current multi-source recommended value for that item; and recording
2 the consistency with which a value provided from a source matches a recommended value.

3
4 The method can further comprise managing parameters of the reference data utility.
5 Configuration management of the reference data utility can comprise managing at least one
6 of: a number of maximum allowable parallel data enhancement processes, types of single-
7 source cleansing processes applied during a data enhancement process, types of cross-source
8 processes applied during a data enhancement process, rules to be applied during specific
9 single-source cleansing processes, and rules to be applied during specific cross-source
10 processes.

11
12 Generating output can comprise: receiving at least one request from a recipient; parsing the at
13 least one request to extract a request specification; and initiating at least one work flow to
14 provide the output data to the recipient.

15
16 The method can comprise providing value added services including at least one service
17 selected from the group consisting of data-driven value added computational functions based
18 on dynamically delivered input datasets, storage and retrieval of business documents, rule-
19 based validation of the applicability of stored business documents to a business transaction
20 and choreography of reference data associated with a business document in support of a
21 business transaction.

22
23 Preferably, the method further comprises maintaining chronological accuracy within the data
24 flow across components of the reference data utility, as well as maintaining a record of total
25 usages by source for each recipient. A report on at least one of source usage and quality of
26 source for each recipient can be generated.

27
28 The method can further comprise creating a market for value added computational services
29 by: establishing a registry for the available services; accepting requests from recipients to
30 execute an identified service with input data provided an on demand dataset; invoking the
31 requested service; returning results from the service computation to the requesting recipient
32 using an on demand dataset; and monitoring service instances to record reporting
33 information. Establishing a registry of available services can comprise: providing a

1 description of the service based on information from a service source, a specification of
2 reference data inputs required to use the service, specification of the outputs generated by
3 each service computation, and maintaining entitlement information from the service origin
4 identifying recipients entitled to use the service.

5
6 Recipient requests for an added value service instance can be handled by receiving an
7 identification of requested service, specification of input reference data used with the service,
8 and delivery specification indicating how output from the service is returned to a client.
9 Invoking a requested service can comprise: validating recipient entitlement to use the service;
10 collecting recipient specified input data by forming and executing an on-demand dataset
11 request to a delivery subsystem based on a transformation of the original request for service
12 execution; verifying that recipient input data meets service input requirements; and executing
13 a service instance.

14
15 Business documents can be stored with annotations relating their content to reference data
16 values. The method can further comprise accepting documents from at least one recipient
17 with reference data annotations, storing annotated documents in the repository, and provide
18 services to recipients based on information arriving from a source relating to the annotations.
19 A validation test can be performed on current values of at least one of unprocessed reference
20 data and processed reference data. The validation test can be performed on request from a
21 recipient.

22
23 The invention is also directed to a computer usable medium having computer readable
24 program code means embodied therein, for causing a computer to effect any of the methods
25 described above and below, herein. The invention is further directed to any data processing
26 apparatus utilizing such computer usable medium.

27 28 B. Data Repository

29 An aspect of the invention is directed to a multi-source multi-tenant data repository of data
30 useful to requestors, comprising: a database management system, the system managing a data
31 base including information elements from a plurality of sources, describing at least one
32 referred entity; annotations of sourcing information for the elements, and entitlements of
33 requesters to information elements based on the sourcing information; data input apparatus

1 for placing the information elements in the database; and data output apparatus for providing
2 data output based on the information elements in response to a request from a requestor, the
3 output containing information to which the requestor is entitled based on the entitlements.

4 The requestors are selected from the group comprising: tenants, clients, sources of reference
5 data, an agent acting on behalf of the repository; software programs acting on behalf of a
6 tenant; and software programs acting on behalf of the repository. The repository further can
7 comprise means for authenticating a requestor as authorized to obtain data from said
8 repository.

9
10 The entitlements comprise at least one of: entitlements to data provided by a specific data
11 source; entitlements to data items generated by a particular value enhancement process; and
12 entitlements to data from a particular process, if all values used to derive the data were from
13 data sources to which a requestor is entitled.

14
15 The repository may comprise a database portion for storing information concerning a set of
16 entitlements governing outputs which a requestor associated with a tenant of the repository is
17 entitled to receive, based on tenants contracting independently with data sources.

18
19 The information elements can comprise element types taken from a list of element types
20 comprising: item instances; versioned attributes; properties; attribute values; business
21 documents; function definitions; rule sets; operational information; media documents;
22 descriptions of standards; hierarchical business data; news information; structured
23 documents; configuration information; and program code.

24
25 A repository entity can comprise at least one item instance information element, each item
26 instance element comprising at least one versioned attribute information element. A
27 repository entity also can comprise at least one property used to select the repository entity in
28 information element selection operations. An item instance may comprise at least one
29 property used to select the item instance in information element selection operations.

30
31 The repository can receive from at least one item instance, process item instances or metadata
32 for inclusion in the repository, wherein the item instance process is uniquely identified by the
33 repository. The repository further can comprise: means for forming at least one repository

entity corresponding to at least one referred entity; and means for associating the item instances or metadata for inclusion with at least one of the repository entities, based on reference to the at least one referred entity.

The database management system can comprise: means for storing in the database as at least one property of at least one item instance, information identifying an item instance process yielding the item instance; means for storing in the database in the form of at least one evolutionarily tracked source data tag at least one event in a history of the item instance, the at least one event comprised of temporal information, sources and agents of the at least one event as received from at least one item instance process; means for storing in the form of at least one evolutionarily tracked source data tag event in a history of versioned attributes of the item instance, the at least one event comprised of temporal information, sources and agents of the at least one event as received from an item instance process; means for storing metadata for each the item instance; means for storing metadata for each repository entity; and means for storing metadata for each the versioned attribute.

The base management system can comprise at least one of: means for storing in at least one evolutionarily tracked source data tag at least one event reflecting operations involving item instances, the at least one event comprised of temporal information, sources and agents of the at least one event; means for creating new item instances; means for affecting existing item instances; means for providing composite processing of a plurality of item instances; means for supplying an item instance with attribute values for a repository entity received from a source dataset; means for supplying an item instance with attribute values for a repository entity resulting from quality assurance and enhancement of values received from a single source dataset; and means for supplying an item instance with attribute values for a repository entity resulting from comparison and selection between values from a plurality of source datasets relating to the same attribute of the same referred entity.

The database management system can also comprise: means for interpreting a request to produce at least one request specification; means for selecting requested information elements in the repository based on the request specification to form a return dataset; means for filtering the return dataset to include only the information elements to which the requester

1 is entitled to form a filtered return dataset; and means for supplying the filtered return dataset
2 to the data output apparatus.

3
4 The requester can be from a group of requesters comprising: an agent acting on behalf of a
5 repository tenant, an agent acting on behalf of the repository; software programs acting on
6 behalf of a repository tenant; and software programs acting on behalf of the repository.

7
8 The means for interpreting can yield a request specification comprising a plurality of aspects
9 taken from a group of aspects comprising: identification of the requester; a selection
10 predicate governing selection of the information elements to be returned; and a sourcing
11 preference comprising prioritization of the at least one preference of a requester when
12 multiple information elements from separate sources are available satisfying the information
13 element selection predicate.

14
15 The means for selecting can comprises at least one taken from a group comprising: means for
16 selecting any repository entity matching the request specification; means for selecting any
17 item instance belonging to the repository entities matching the request specification; means
18 for selecting any attribute value forming the item instances matching the request
19 specification; means for selecting any metadata associated with the any repository entity
20 matching the request specification; means for selecting any metadata associated with the any
21 item instance matching the request specification; means for selecting any metadata associated
22 with the any attribute value matching the request specification; and means for applying
23 specified requester sourcing preference to form a return dataset comprising: the item
24 instances, the attribute values belonging to the item instances, and the metadata.

25
26 The means for filtering can comprises at least one from a group comprising: means for
27 ensuring the entitlement of the at least one requester to any item instance process responsible
28 for generating any of the item instances selected; means for ensuring the entitlement of the at
29 least one requester to any attribute value selected, wherein the requester is entitled to receive
30 the any attribute value only if the requester is entitled to at least one source providing the
31 attribute value; and means for removing from the return dataset any information element to
32 which the requester is not entitled.

1 The invention is also directed to an information processing apparatus including means for
2 maintaining a multi-source multi-tenant data repository, wherein the means for maintaining
3 comprises: means for forming the multi-source multi-tenant data repository to include
4 information elements from a plurality of sources, describing at least one referred entity;
5 means for annotating a plurality of elements from the information elements in the multi-
6 source multi-tenant data repository with sourcing information; and means for maintaining
7 information about entitlement of requesters to information elements based on the sourcing
8 information.

9
10 The invention is further directed to a multi-source multi-tenant reference data repository,
11 comprising: means for inputting into the repository information about business instrument
12 information, corporate hierarchy information, corporate news information and other
13 information from multiple sources; means for annotating changes to the business instrument
14 information, corporate hierarchy information, corporate news information, and other
15 information, and sources of each change; means for maintaining information about
16 entitlement of requesters to obtain the business instrument information, corporate hierarchy
17 information, corporate news information, and other information, based on entitlement to the
18 sources and processes which have provided or modified this information; and means for
19 responding to at least one request from at least one requester to return at least a subset of the
20 business instrument information, corporate hierarchy information, corporate news
21 information, and other information, based on requester-specified selection and sourcing
22 preferences and subject to the entitlements.

23
24 The invention also encompasses a repository apparatus comprising: means for forming the
25 repository with information element structures in place; means for inserting arriving
26 information elements into a store of the repository; means for annotating each information
27 element with annotations describing evolutionary history of the each information element in
28 forming an evolutionarily tracked source data tag for the each information element; means for
29 maintaining source based entitlement information for authorized requesters of repository
30 information and data sources to which each authorized requester is entitled; and means for
31 employing the evolutionarily tracked source data tag together with source based entitlements
32 in a process providing entitlement enforced controlled access to information included in the
33 repository.

1 The invention is also directed to a method for maintaining a multi-source multi-tenant data
2 repository of data useful to requestors, comprising: forming the multi-source multi-tenant
3 data repository to include information elements from a plurality of sources, describing at least
4 one referred entity; annotating a plurality of the elements with sourcing information; and
5 maintaining information about entitlement of requesters to information elements based on the
6 sourcing information.

7
8 The entitlements comprise at least one of: entitlements to data provided by a specific data
9 source; entitlements to data items generated by a particular value enhancement process; and
10 entitlements to data from a particular process, if all values used to derive the data were from
11 data sources to which a requestor is entitled. Tenants of the repository contract
12 independently with data sources, and data enhancement process owners to
13 determine a set of entitlements governing data which requestors associated with tenants are
14 entitled to receive.

15
16 The method can further comprise responding to at least one request from at least one
17 requester to return a set of information elements based on requester-specified selection
18 predicates and sourcing preferences and subject to the entitlement of the at least one
19 requester. The step of responding can comprise: receiving at least one request from at least
20 one requester; interpreting the at least one request to produce at least one request
21 specification; selecting requested information elements in the multi-source multi-tenant data
22 repository based on the request specification to form a return dataset; filtering the return
23 dataset include only the information elements to which the requester is entitled to form a
24 filtered return dataset; and returning the filtered return dataset to the requester.

25
26 The step of interpreting can yield a request specification comprising a plurality of aspects
27 taken from a group of aspects comprising: identification of the requester; a selection
28 predicate governing selection of the information elements to be returned; and a sourcing
29 preference comprising prioritization of the at least one requester's preference when multiple
30 information elements from separate sources are available satisfying the information element
31 selection predicate.

1 The step of selecting can comprise at least one step taken from a group comprising: selecting
2 any repository entity matching the request specification; selecting any item instance
3 belonging to the repository entities matching the request specification; selecting any attribute
4 value forming the item instances matching the request specification; selecting any metadata
5 associated with the any repository entity matching the request specification; selecting any
6 metadata associated with the any item instance matching the request specification; selecting
7 any metadata associated with the any attribute value matching the request specification; and
8 applying specified requester sourcing preference to form a return dataset comprising: the item
9 instances, the attribute values belonging to the item instances, and the metadata.

11 The step of filtering can comprise at least one step taken from a group of steps comprising:
12 ensuring the entitlement of the at least one requester to any item instance process responsible
13 for generating any of the item instances selected; ensuring the entitlement of the at least one
14 requester to any attribute value selected, wherein the requester is entitled to receive the any
15 attribute value only if the requester is entitled to at least one source providing the attribute
16 value; and removing from the return dataset any information element to which the requester
17 is not entitled.

19 In the method, the multi-source multi-tenant data repository can comprise a plurality of
20 information element types taken from a list of information element types comprising: item
21 instances; versioned attributes; properties; attribute values; business documents; function
22 definitions; rule sets; operational information; media documents; descriptions of standards;
23 hierarchical business data; news information; structured documents; configuration
24 information; and program code.

26 In accordance with the method, a repository entity may have at least one item instance
27 information element, each item instance comprising at least one versioned attribute
28 information element. A repository entity may comprise at least one property used to select the
29 repository entity in information element selection operations. An item instance may
30 comprise at least one property used to select the item instance in information element
31 selection operations.

1 The forming the multi-source multi-tenant data repository can comprises at least one step
2 taken from a group of steps comprising: the multi-source multi-tenant data repository
3 receiving from at least one item instance process item instances or metadata for inclusion in
4 the repository, wherein the item instance process is uniquely identified by the repository; the
5 repository forming at least one repository entity corresponding to at least one referred entity;
6 and the repository associating the item instances or metadata for inclusion with at least one of
7 the repository entities based on reference to the at least one referred entity.

8
9 The annotating can comprise at least one step taken from a group of steps comprising: storing
10 as at least one property of at least one item instance, information identifying-an item instance
11 process yielding the item instance; storing in the form of at least one evolutionarily tracked
12 source data tag at least one event in a history of the item instance, the at least one event
13 comprised of temporal information, sources and agents of the at least one event as received
14 from at least one item instance process; storing in the form of at least one evolutionarily
15 tracked source data tag event in a history of versioned attributes of the item instance, the at
16 least one event comprised of temporal information, sources and agents of the at least one
17 event as received from an item instance process; storing metadata for each the item instance;
18 storing metadata for each repository entity; and storing metadata for each the versioned
19 attribute.

20
21 The repository item instance process may comprise at least one step taken from a group of
22 steps comprising: storing in at least one evolutionarily tracked source data tag at least one
23 event reflecting operations involving item instances, the at least one event comprised of
24 temporal information; sources and agents of the at least one event; employing a process that
25 creates new item instances; employing a process that affects existing item instances;
26 employing a composite process comprised of a plurality of item instance processes; supplying
27 an item instance with attribute values for a repository entity received from a source dataset;
28 supplying an item instance with attribute values for a repository entity resulting from quality
29 assurance and enhancement of values received from a single source dataset; and supplying an
30 item instance with attribute values for a repository entity resulting from comparison and
31 selection between values from a plurality of source datasets relating to the same attribute of
32 the same referred entity.

1 In the method the requester can be from a group of requesters comprising: an agent acting on
2 behalf of a repository tenant, an agent acting on behalf of the repository; software programs
3 acting on behalf of a repository tenant; and software programs acting on behalf of the
4 repository.

5
6 The invention is also directed to a method for maintaining a multi-source multi-tenant
7 reference data repository, comprising: forming the multi-source multi-tenant reference data
8 repository to include information about business instrument information, corporate hierarchy
9 information, corporate news information and other information from multiple sources;
10 annotating changes to the business instrument information, corporate hierarchy information,
11 corporate news information, and other information, and sources of each change; maintaining
12 information about entitlement of requesters to obtain the business instrument information,
13 corporate hierarchy information, corporate news information, and other information, based on
14 entitlement to the sources and processes which have provided or modified this information;
15 and responding to at least one request from at least one requester to return at least a subset of
16 the business instrument information, corporate hierarchy information, corporate news
17 information, and other information, based on requester-specified selection and sourcing
18 preferences and subject to the entitlements. The at least one requester can be a client of at
19 least one service provider. In accordance with the method, the business instrument
20 information can be financial information.

21
22 In accordance with the method, the source can be from a group of sources comprising:
23 providers of business data; providers of financial data; providers of ratings data; providers of
24 corporate news data; and providers of corporate hierarchy data. The financial information
25 can include data taken from a group of data comprising at least one of: stock instruments;
26 bond instruments; derivative instruments; fixed income instruments; mortgage-backed
27 instruments; counterparty information; financial transaction information; corporate events
28 information; financial trade information; financial settlement information; financial
29 documents; other financial instruments; and other financial data.

30
31 In the method, the forming of the multi-source multi-tenant reference data repository can
32 comprise keeping distinct versions of historical states of the business instrument information,
33 corporate hierarchy information, corporate news information and other information.

1 The method is also directed to a data repository method comprising: forming the repository
2 with information element structures in place; inserting arriving information elements into a
3 store of the repository; annotating each information element with annotations describing
4 evolutionary history of the each information element in forming an evolutionarily tracked
5 source data tag for the each information element; maintaining source based entitlement
6 information for authorized requesters of repository information and data sources to which
7 each authorized requester is entitled; and employing the evolutionarily tracked source data
8 tag together with source based entitlements in a process providing entitlement enforced
9 controlled access to information included in the repository.

10
11 The method can further comprise retrieval requests arriving as input; and the process of
12 providing can comprise forming retrieval responses returned as output for the processing.

13
14 The method can further comprise employing the repository to store other items that reside in
15 a data store, the other items comprising at least one of: business value added functions,
16 business documents, functional rule sets; operational rule sets, functional log records, and
17 operational log records.

18
19 The method can further comprising associating the evolutionarily tracked source data tags
20 with at least one information element in the repository; and the annotating can comprise
21 recording sourcing information to provide traceability to sources that contributed to creation
22 of an information element value. The maintaining can comprise updating entitlement
23 information received as input. A step in maintaining can be performed by a service provider.

24
25 The invention is also directed to a computer usable medium having computer readable
26 program code means embodied therein, the computer readable program code means being for
27 causing a computer to effect any or all of the methods mentioned above and described below.

28
29 The invention may be used with the multi-source multi-tenant reference data utility described
30 herein, for delivering high quality reference data in response to requests from clients,
31 implemented using a shared infrastructure, and also providing added value services using the
32 client's reference data. Data cleansing and quality assurance of the received data with full
33 tracking of the sourcing of each value, storage of resulting entity values in a repository which

allows retrievals and enforces source based entitlements, and delivery of retrieved data in the form of on demand datasets supporting a wide range of client application needs, may be utilized. An advantageous implementation has additional services for reporting on data quality and usage, a selection of value adding data driven computations and business document storage. By using a shared infrastructure and amortizing the costs of data quality assurance across a plurality of clients, while ensuring that clients only receive values from data sources to which they are licensed, better quality data is delivered at lower cost than other methods currently available.

C. Data Value Enhancement

An aspect of the invention is directed to a method for enhancing the value of reference data, comprising: subjecting the data to at least one value enhancing process; and maintaining a complete record of all sources of the data and all enhancement processing steps contributing to the generation of each enhanced element of the reference data. The method can further comprise receiving data concerning a referred item from a first data source; and generating enhanced values based on comparing and processing values for the same referred item from multiple sources. In addition the method generally comprises performing at least one of: validating the data by at least one of a manual process and an automatic process; normalizing the data by at least one of a manual process and an automatic process; and cleansing the data by at least one of a manual process and an automatic process.

Generally the reference data includes source elements, and the validating comprises: obtaining the at least one source element from a source description; and performing at least one step taken from a group of steps comprising: detecting any source element which does not conform to the source description; flagging any source element which does not conform to the source description; correcting any source element which does not conform to the source description; and removing any source element which does not conform to the source description; and recording to at least one evolutionarily tracked sourced data tag any event generated by the step of performing validation.

The normalizing comprises: obtaining the source element in a source description; converting the source element based on the source description to at least one target information element based on a corresponding target description, wherein the target description is information

describing structure, contents and constraints of repository information elements, as they are stored in a repository; and performing at least one step taken from a group of steps comprising: detecting any source element which cannot be normalized; flagging any source element which cannot be normalized; correcting any source element which cannot be normalized; removing any source element which cannot be normalized; and recording to at least one evolutionarily tracked sourced data tag any event generated by the step of performing normalization.

The cleansing comprises at least one of: automated execution of at least one rule from at least one rule set containing source-specific cleansing rules; examination of the source element values by one skilled in subject matter relevant to at least one referred entity; application of any rule from the at least one rule set containing source-specific rules by one skilled in subject matter relevant to at least one referred entity; removal of any of the source element values; augmentation of any of the source element values; correction of any of the source element values; annotation of any quality concerns; reporting back to the source, inquiries regarding quality of the source element in question; and recording any event generated by any action, taken from the group of actions, to at least one evolutionarily tracked sourced data tag.

Advantageously, the method comprises: selecting all of the source elements that contain information describing a same referred entity; applying predetermined rules to at least one of the source elements and attributes of the elements; selecting one of a preferred or recommended item from the alternatives provided by the different sources by at least one of: creating at least one new item based on a combination of attributes provided by the different sources, or modifying the elements provided by the different sources; creating a new corresponding evolutionarily tracked source data tag when at least one new item or items is created; and annotating the evolutionarily tracked source data tag at the source item level with the information about the cross-source processing applied to the item.

If an existing element was selected but no attributes were modified, the method further comprises providing an annotation at the item level to denote which parent sources matched the selection made. If either modification of data at an attribute level or a creation of a new item occurs, the method further comprises separately annotating an exact set of sources for each attribute.

1 The invention is also directed to a data processing method comprising producing at least one
2 evolutionarily tracked source tagged dataset, comprising: receiving at least one source-dataset
3 from at least one source, wherein a source element includes one of a source attribute and a
4 source item, each source-dataset having at least one source item, each source item having at
5 least one source attribute; recording a source identification for each source element, and a
6 source identification for each source-dataset in at least one evolutionarily tracked source data
7 tag; obtaining relevant information resulting from the step of receiving and the step of
8 recording to form at least one recordable event in at least one evolutionarily tracked source
9 data tag; and forming the at least one evolutionarily tracked source tagged dataset to include
10 at least one evolutionarily tracked source data tag, the at least one evolutionarily tracked
11 source data tag including the at least one recordable event, and including at least one source
12 of the at least one recordable event.

13
14 The method can further comprise: invoking at least one rule from at least one rule-set on at
15 least one of the source dataset, the source element, and an information element; and
16 obtaining relevant information evolving from the step of invoking to form at least one other
17 recordable event in at least one evolutionarily tracked source data tag.

18
19 The at least one rule set can comprise at least one rule taken from a group of rules,
20 comprising: rules for checking range tolerance of source attribute values; rules for checking
21 rate of change of source attribute values; rules for checking consistency of source attribute
22 values with other relevant source attribute values; rules for checking structural consistency of
23 source elements; rules for checking consistency of source elements with other relevant source
24 elements; rules for checking suitability of source elements for transformation into target
25 information elements within a multi-source multi-tenant data repository, as described by a
26 target description; rules for checking compatibility of source element values with existing
27 referred entity information; rules for identifying source elements as having come from a
28 particular source; rules for comparing source elements in the context of a specific cross-
29 source process; rules applicable to source datasets; rules applicable to source elements; and
30 rules applicable to information elements. The at least one rule is grouped into at least one rule
31 set according to applicability of the at least one rule to at least one processing stage taken
32 from a group of processing stages, comprising: validation; normalization; source-specific
33 cleansing; and a cross-source process.

1 A rule can comprise at least one of: an executable test condition; a correction method;
2 information identifying the at least one rule set to which the rule belongs.

3
4 In accordance with the method a recordable event can includes data taken from a group of
5 data comprising: an event description; an agent of the event; temporal information associated
6 with the event; at least one source of the event; an identifier of the event; information
7 required to correlate the event with the information element to which it applies; and a
8 classification of the event.

9
10 The step of invoking can comprise at least one step taken from a group of steps comprising:
11 performing validation on at least one source element; performing normalization on the at
12 least one source element; performing source-specific cleansing on the at least one source
13 element; and executing at least one cross-source process on the at least one source element.

14
15 The step of performing validation on the at least one source element can comprise: obtaining
16 the at least one source element from a source description; and performing at least one step
17 taken from a group of steps comprising: detecting any source element which does not
18 conform to the source description; flagging any source element which does not conform to
19 the source description; correcting any source element which does not conform to the source
20 description; and removing any source element which does not conform to the source
21 description; and recording to at least one evolutionarily tracked sourced data tag any event
22 generated by the step of performing validation.

23
24 The step of performing normalization on the at least one source element can comprise:
25 obtaining the source element in a source description; converting the source element based on
26 the source description to at least one target information element based on a corresponding
27 target description, wherein the target description is information describing structure, contents
28 and constraints of repository information elements, as they are stored in a repository; and
29 performing at least one step taken from a group of steps comprising: detecting any source
30 element which cannot be normalized; flagging any source element which cannot be
31 normalized; correcting any source element which cannot be normalized; removing any source
32 element which cannot be normalized; and recording to at least one evolutionarily tracked
33 sourced data tag any event generated by the step of performing normalization.

1 The step of performing source-specific cleansing can comprise an action taken from a group
2 of actions comprising: automated execution of the at least one rule from the at least one rule
3 set containing source-specific cleansing rules; examination of the source element values by
4 one skilled in subject matter relevant to at least one referred entity; application of any rule
5 from the at least one rule set containing source-specific rules by one skilled in subject matter
6 relevant to at least one referred entity; removal of any of the source element values;
7 augmentation of any of the source element values; correction of any of the source element
8 values; annotation of any quality concerns; reporting back to the source, inquiries regarding
9 quality of the source element in question; and recording any event generated by any action,
10 taken from the group of actions, to at least one evolutionarily tracked sourced data tag.

11
12 The step of executing at least one cross-source process can comprise an action taken from a
13 group of actions comprising: examining source elements from a plurality of data sources
14 referring to a same referred entity; automatically executing at least one rule from the at least
15 one rule set including cross-source process rules specific to the at least one cross-source
16 process; examining the source elements by one skilled in subject matter relevant to the same
17 referred entity; applying any rule from the at least one rule set containing cross-source
18 process rules specific to the at least one cross-source process by one skilled in such subject
19 matter; selecting any of the source elements values as a preferred value; comparing any of the
20 source elements; removing any of the source element values; augmenting any of the source
21 element values; modifying any of the source element values; annotating any quality concerns;
22 creating at least one item instance to include results of the at least one cross-source process;
23 modifying at least one item instance to include the results of the at least one cross-source
24 process; adding identification information to at least one item instance to recognize the at
25 least one item instance as target of the at least one cross-source process; and recording any
26 event generated by any action, taken from the group of actions, to at least one evolutionarily
27 tracked sourced data tag.

28
29 The method further can comprising resolving differences detected during the step of
30 comparing the source elements through at least one step taken from a group of steps
31 comprising: automatically selecting source elements based on business rules; automatically
32 selecting source elements based on algorithms; manually selecting a recommended source
33 element by one skilled in the subject, based on knowledge of the subject area; manually

1 selecting a recommended source element by one skilled in the subject, based on freely
2 available public information; manually creating a recommended source element by one
3 skilled in the subject, based on knowledge of the subject area; manually creating a
4 recommended source element by one skilled in the subject, based on freely available public
5 information; and recording any event generated by any step taken from the group of steps, to
6 at least one evolutionarily tracked sourced data tag.

7
8 The step of recording can comprise identifying which sources matched a selected preferred
9 source element value. In addition, the method can further comprise: presenting the at least
10 one source element to one skilled in such subject; enabling performance of manual validation
11 of the at least one source element; performing manual validation; and recording to at least
12 one evolutionarily tracked sourced data tag any event generated by the step of performing
13 manual normalization.

14
15 The method also may further comprise: presenting the at least one source element to one
16 skilled in such subject; enabling performance of manual normalization of the at least one
17 source element; performing manual normalization; and recording to at least one
18 evolutionarily tracked sourced data tag any event generated by the step of performing manual
19 normalization.

20
21 An overall set of reference data being processed can be on a variety of distinct topics, with
22 the source datasets of reference data being individually cleansed, each source supplying
23 source items on at least one topic.

24
25 The invention is also directed to a quality assurance process for reference data, comprising:
26 receiving reference data in a source dataset from at least one source, each source-dataset
27 having at least one source item, each source item having at least one source attribute, wherein
28 a source element is one of a source item and a source attribute; recording a source
29 identification for each source element, and a source identification for each source-dataset in
30 at least one evolutionarily tracked source data tag, such that at least one evolutionarily
31 tracked source data tag is associated with each source element; recording data evolution
32 events from steps of validating, normalizing, single-source processing, and cross-source
33 processing, of source elements in the at least one evolutionarily tracked source data tag; and

1 forming the at least one evolutionarily tracked source tagged dataset to include at least one
2 evolutionarily tracked source data tag, the at least one evolutionarily tracked source data tag
3 including the at least one data evolution event and a source of the at least one data evolution
4 event.

5
6 The invention is further directed to an article of manufacture comprising a computer usable
7 medium having computer readable program code means embodied therein for causing data
8 processing, the computer readable program code means in the article of manufacture
9 comprising computer readable program code means for causing a computer to effect any one
10 of the methods mentioned above and described in more detail below.

11
12 In accordance with yet another aspect, the invention is directed to apparatus for enhancing the
13 value of reference data, comprising: means for subjecting the data to at least one value
14 enhancing process; and a database for maintaining a complete record of all sources of the
15 data and all enhancement processing steps contributing to the generation of each enhanced
16 element of the reference data. The apparatus can further comprise: means for receiving data
17 concerning a referred item from a first data source; and means for generating enhanced values
18 based on comparing and processing values for the same referred item from multiple sources.

19
20 The apparatus can further comprising at least one of: validating means for validating the data
21 by at least one of a manual process and an automatic process; normalizing means for
22 normalizing the data by at least one of a manual process and an automatic process; and
23 cleansing means for cleansing the data by at least one of a manual process and an automatic
24 process.

25
26 Generally the reference data includes source elements, and the validating means comprises:
27 means for obtaining the at least one source element from a source description; and means for
28 performing at least one step taken from a group of steps comprising: detecting any source
29 element which does not conform to the source description; flagging any source element
30 which does not conform to the source description; correcting any source element which does
31 not conform to the source description; and removing any source element which does not
32 conform to the source description; and means for recording to at least one evolutionarily
33 tracked sourced data tag any event generated by the step of performing validation.

1 The means for normalizing can comprise: means for obtaining the source element in a source
2 description; means for converting the source element based on the source description to at
3 least one target information element based on a corresponding target description, wherein the
4 target description is information describing structure, contents and constraints of repository
5 information elements, as they are stored in a repository; and means for performing at least
6 one step taken from a group of steps comprising: detecting any source element which cannot
7 be normalized; flagging any source element which cannot be normalized; correcting any
8 source element which cannot be normalized; means for removing any source element which
9 cannot be normalized; and means for recording to at least one evolutionarily tracked sourced
10 data tag any event generated by the step of performing normalization.

11
12 The cleansing means comprises at least one of: means for automated execution of at least one
13 rule from at least one rule set containing source-specific cleansing rules; means for
14 examination of the source element values by one skilled in subject matter relevant to at least
15 one referred entity; means for application of any rule from the at least one rule set containing
16 source-specific rules by one skilled in subject matter relevant to at least one referred entity;
17 means for removal of any of the source element values; means for augmentation of any of the
18 source element values; means for correction of any of the source element values; means for
19 annotation of any quality concerns; means for reporting back to the source, inquiries
20 regarding quality of the source element in question; and means for recording any event
21 generated by any action, taken from the group of actions, to at least one evolutionarily
22 tracked sourced data tag.

23
24 The apparatus can further comprising means for receiving the reference data from multiple
25 sources, and means for selecting and enhancing the data by at least one of a manual process
26 and an automatic process to produce data of enhanced value.

27
28 The apparatus can comprise: means for selecting all of the source elements that contain
29 information describing a same referred entity; means for applying predetermined rules to at
30 least one of the source elements and attributes of the elements; means for selecting one of a
31 preferred or recommended item from the alternatives provided by the different sources by at
32 least one of: creating at least one new item based on a combination of attributes provided by
33 the different sources, or modifying the elements provided by the different sources; means for

1 creating a new corresponding evolutionarily tracked source data tag when at least one new
2 item or items is created; and means for annotating the evolutionarily tracked source data tag
3 at the source item level with the information about the cross-source processing applied to the
4 item.

5
6 The apparatus can further comprising means for providing an annotation at the item level to
7 denote which parent sources matched the selection made, if an existing element was selected
8 but no attributes were modified. The apparatus also can further comprising means for
9 separately annotating an exact set of sources for each attribute, if either modification of data
10 at an attribute level or a creation of a new item occurs.

11
12 According to yet another aspect, the invention is directed to a data processing apparatus for
13 producing at least one evolutionarily tracked source tagged dataset, comprising: at least one
14 input for receiving at least one source-dataset from at least one source, each source-dataset
15 having at least one source item, each source item having at least one source attribute; memory
16 for recording a source identification for each source attribute, a source identification for each
17 source item, and a source identification for each source-dataset; apparatus for invoking at
18 least one rule from at least one rule-set on at least one of: the source-dataset, the source item,
19 and the attribute; apparatus for retaining relevant information about the steps of invoking,
20 receiving and recording resulting in at least one recordable event; a processor for forming the
21 at least one evolutionarily tracked source tagged dataset to include the at least one recordable
22 event and an event originator of the at least one recordable event.

23
24 In accordance with the invention, a data processing apparatus for assuring quality of
25 reference data, comprises: means for receiving reference data in a source dataset from at least
26 one source, each source-dataset having at least one source item, each source item having at
27 least one source attribute, wherein a source element is one of a source item and a source
28 attribute; means for recording a source identification for each source element, and a source
29 identification for each source-dataset in at least one evolutionarily tracked source data tag,
30 such that at least one evolutionarily tracked source data tag is associated with each source
31 element; means for recording data evolution events from steps of validating, normalizing,
32 single-source processing, and cross-source processing, of source elements in the at least one
33 evolutionarily tracked source data tag; and means for forming the at least one evolutionarily

1 tracked source tagged dataset to include at least one evolutionarily tracked source data tag,
2 the at least one evolutionarily tracked source data tag including the at least one data evolution
3 event and a source of the at least one data evolution event.
4

5 The invention may be used with a multi-source multi-tenant reference data utility, as
6 described herein, for delivering high quality reference data in response to requests from
7 clients, implemented using a shared infrastructure, and also providing added value services
8 using the client's reference data. Data cleansing and quality assurance of the received data
9 with full tracking of the sourcing of each value, storage of resulting entity values in a
10 repository which allows retrievals and enforces source based entitlements, and delivery of
11 retrieved data in the form of on demand datasets supporting a wide range of client application
12 needs, may be utilized. An advantageous implementation has additional services for reporting
13 on data quality and usage, a selection of value adding data driven computations and business
14 document storage. By using a shared infrastructure and amortizing the costs of data quality
15 assurance across a plurality of clients, while ensuring that clients only receive values from
16 data sources to which they are licensed, better quality data is delivered at lower cost than
17 other methods currently available.
18

19 D. On-Demand Data Delivery

20 Another aspect of the invention is directed to an information delivery method for satisfying at
21 least one on-demand dataset request, comprising: processing the at least one on-demand
22 dataset request from at least one requester; producing at least one parsed on-demand dataset
23 request specification; configuring at least one on demand dataset production process to
24 produce at least one on-demand dataset satisfying the at least one on-demand dataset request;
25 and executing the at least one on demand dataset production process to return the at least one
26 on-demand dataset to the at least one requester; wherein the on-demand dataset is limited to
27 data derived from sources and data enhancements to which the requester is entitled.
28

29 The on demand dataset request can comprise at least one on demand dataset request
30 specification enabling the requester to specify characteristics taken from a group of
31 characteristics comprising: information items to be returned; selection of information items to
32 be returned; sourcing preference to select between alternate available values; delivery mode;
33 delivery timing; transport protocol; transport protocol ports; security tokens; preferred data

1 format; data transformation rules; custom functions to be invoked; custom filtering rules;
2 exception handling instructions; annotation instructions; data delivery feedback mechanism
3 instructions; delivery endpoint; delivery intermediaries; metadata handling instructions;
4 logging instructions; routing instructions; data merging instructions; and data splitting
5 instructions. The method can further comprise receiving information assembled in an on
6 demand dataset from a multi-source multi-tenant data repository.

7
8 The at least one requester can be taken from a group of requesters comprising: a tenant of a
9 multi-source multi-tenant repository; an agent acting on behalf of the tenant; an agent acting
10 on behalf of the repository; a program acting on behalf of the repository; and a program
11 acting on behalf of the tenant.

12
13 The method can further comprising using a delivery mode to deliver the at least one on
14 demand dataset, the delivery mode including at least one delivery modality from a group of
15 delivery modalities comprising: quasi real time delivery; scheduled batch delivery; datamart
16 delivery; one time query delivery; e-mail delivery; fax delivery; online delivery; printed hard
17 copy delivery; automated voice delivery; magnetic tape delivery; optical disc delivery; digital
18 media delivery; video delivery; and condition triggered delivery.

19
20 The configuring of the at least one on demand dataset production process can be enabled by
21 employing at least one activity building block for separable steps of the at least one on
22 demand dataset production process.

23
24 The at least one activity building block can be taken from a collection of activity building
25 blocks wherein each block enables at least one function taken from a group of functions
26 comprising: information element selection; sourcing selection; entitlement enforcement; data
27 assembly; delivery scheduling; transfer protocol handling; standard format transformation;
28 database load; custom data transformation; logging; function execution; filtering; annotating;
29 routing; data splitting; data merging; and data transmission.

30
31 The configuring of at least one on demand dataset production process can comprise:
32 employing the at least one parsed on demand dataset request specification; selecting at least
33 one activity building block for inclusion in the on demand dataset production process,

wherein the at least one activity building block satisfies at least one characteristic of the at least one parsed on demand dataset request specification; parameterizing any selected activity building block with execution parameters; and assembling selected parameterized activity blocks into the at least one on demand dataset production process.

The executing of the on demand dataset production process can include at least one step taken from a group of steps comprising: executing logic of an assembly flow included within the on demand dataset production process; executing each parameterized activity building block of the on demand dataset production process as many times as indicated by the logic; transmitting the at least one on demand dataset to the at least one requester; recording aspects of actions taken in response to the at least one request to enable repetition of the step of executing at a later time; and logging at least one aspect of delivery taken from a group of aspects. The aspects can include: time of delivery; date of delivery; contents of delivery; requester of the delivery; mode of the delivery; size of the delivery; execution time of the delivery process; identifier of the delivery; any error of the delivery process; any warning of the delivery process; success of the delivery process; feedback of the delivery process; non-repudiation information associated with the delivery process; security features of the delivery; and the at least one on demand dataset request.

The on demand dataset request can be initiated by an action taken from a group of actions comprising: a manually initiated request; an automatically initiated request; a one-time request; a data arrival event; a data availability event; a data deletion event; a data change event; a data temporal event; a scheduled request; a request received through an intermediary; and an on-line initiated request.

Significantly, the method is scalable to permit delivery of information via multiple delivery requests from multiple requesters. The method can utilize automated handling of information delivery requests, and can be specifically configured for needs of each delivery request.

The invention is also directed to a method for returning reference data from a multi-source multi-tenant data repository in response to requests from requesters, comprising: receiving at least one request from a requester; parsing the at least one request to extract the request specification; configuring at least one workflow to deliver the requested reference data based

1 on entitlements of the requester, as well as the selection criteria, sourcing preferences and
2 other preferences contained in requester's request; and executing the workflow and delivering
3 the requested reference data to the requester.

4
5 The request specification includes at least one preference taken from a group of preferences
6 comprising: selection criteria; sourcing preferences; data format preferences; delivery
7 transport preferences; and preferences particular to the requester.

8
9 The configuring can include at least one action taken from a group of actions comprising:
10 retrieving the requested reference data; filtering the requested reference data; and formatting
11 the requested reference data.

12
13 The invention is also directed to an article of manufacture comprising a computer usable
14 medium having computer readable program code means embodied therein for causing
15 information processing, the computer readable program code means in the article of
16 manufacture comprising computer readable program code means for causing a computer to
17 effect the any or all of the methods mentioned above and described in more detail below.

18
19 The invention is also directed to an information processing apparatus for satisfying at least
20 one on demand dataset request, comprising: a processor for processing the at least one on
21 demand dataset request from at least one requester; a computer program component
22 executable for producing at least one parsed on demand dataset request specification; a
23 program configuration means for configuring at least one on demand dataset production
24 process to produce at least one on-demand dataset satisfying the at least one on demand
25 dataset request; and computer code for executing the at least one on demand dataset
26 production process to return the at least one on-demand dataset to the at least one requester;
27 wherein the on-demand dataset is limited to data derived from sources and data
28 enhancements to which the requester is entitled.

29
30 The processor processes an on demand dataset request comprises at least one on demand
31 dataset request specification enabling the requester to specify characteristics taken from a
32 group of characteristics comprising: information items to be returned; selection of
33 information items to be returned; sourcing preference to select between alternate available

1 values; delivery mode; delivery timing; transport protocol; transport protocol ports; security
2 tokens; preferred data format; data transformation rules; custom functions to be invoked;
3 custom filtering rules; exception handling instructions; annotation instructions; data delivery
4 feedback mechanism instructions; delivery endpoint; delivery intermediaries; metadata
5 handling instructions; logging instructions; routing instructions; data merging instructions;
6 and data splitting instructions.

7
8 In a preferred embodiment, the apparatus further comprises means for receiving information
9 assembled in an on demand dataset from a multi-source multi-tenant data repository.

10
11 The apparatus can further comprise means for receiving the on demand dataset request from
12 at least one requester taken from a group of requesters comprising: a tenant of a multi-source
13 multi-tenant repository; an agent acting on behalf of the tenant; an agent acting on behalf of
14 the repository; a program acting on behalf of the repository; and a program acting on behalf
15 of the tenant.

16
17 The apparatus can further comprise a delivery component, wherein the delivery component
18 includes at least one delivery modality from a group of delivery modalities comprising: quasi
19 real time delivery; scheduled batch delivery; datamart delivery; one time query delivery; e-
20 mail delivery; fax delivery; online delivery; printed hard copy delivery; automated voice
21 delivery; magnetic tape delivery; optical disc delivery; digital media delivery; video delivery;
22 and condition triggered delivery.

23
24 The apparatus may further comprise at least one activity building block for building separable
25 steps of the at least one on demand dataset production process so as to configure the at least
26 one on demand dataset production process. The at least one activity building block can be
27 taken from a collection of activity building blocks wherein each block enables at least one
28 function taken from a group of functions comprising: information element selection; sourcing
29 selection; entitlement enforcement; data assembly; delivery scheduling; transfer protocol
30 handling; standard format transformation; database load; custom data transformation;
31 logging; function execution; filtering; annotating; routing; data splitting; data merging; and
32 data transmission.

1 The program configuration means can comprise: means for employing the at least one parsed
2 on demand dataset request specification; means for selecting at least one activity building
3 block for inclusion in the on demand dataset production process, wherein the at least one
4 activity building block satisfies at least one characteristic of the at least one parsed on
5 demand dataset request specification; means for parameterizing a selected activity building
6 block with execution parameters; and means for assembling selected parameterized activity
7 blocks into the at least one on demand dataset production process.

8
9 The computer code for executing the on demand dataset production process can include at
10 least one computer code component taken from a group of computer code components,
11 comprising: computer code for executing logic of an assembly flow included within the on
12 demand dataset production process; computer code for executing each parameterized activity
13 building block of the on demand dataset production process as many times as indicated by the
14 logic; computer code for transmitting the at least one on demand dataset to the at least one
15 requester; computer code for recording aspects of actions taken in response to the at least one
16 request to enable repetition of the step of executing at a later time; and computer code for
17 logging at least one aspect of delivery taken from a group of aspects. The aspects can
18 include: time of delivery; date of delivery; contents of delivery; requester of the delivery;
19 mode of the delivery; size of the delivery; execution time of the delivery process; identifier of
20 the delivery; any error of the delivery process; any warning of the delivery process; success
21 of the delivery process; feedback of the delivery process; non-repudiation information
22 associated with the delivery process; security features of the delivery; and the at least one on
23 demand dataset request.

24
25 The apparatus can further comprise means for initiating the on demand dataset request in
26 response to an action taken from a group of actions comprising: a manually initiated request;
27 an automatically initiated request; a one-time request; a data arrival event; a data availability
28 event; a data deletion event; a data change event; a data temporal event; a scheduled request;
29 a request received through an intermediary; and an on-line initiated request.

30
31 The invention is also directed to an apparatus for returning reference data from a multi-source
32 multi-tenant data repository in response to requests from requesters, comprising: means for
33 receiving at least one request from a requester; means for parsing the at least one request to

1 extract the request specification; means for configuring at least one workflow to deliver the
2 requested reference data based on entitlements of the requester, as well as the selection
3 criteria, sourcing preferences and other preferences contained in requester's request; means
4 for executing the workflow and delivering the requested reference data to the requester; and
5 means for limiting the on-demand dataset to data derived from sources and data
6 enhancements to which the requester is entitled.

7
8 The apparatus also can comprise means responsive to the request specification including at
9 least one preference taken from a group of preferences comprising: selection criteria;
10 sourcing preferences; data format preferences; delivery transport preferences; and preferences
11 particular to the requester.

12
13 The means for configuring includes at least one means taken from a group of means
14 comprising: means for retrieving the requested reference data; means for filtering the
15 requested reference data; and means for formatting the requested reference data.

16 The invention is also directed to an apparatus for returning reference data from a multi-source
17 multi-tenant data repository in response to requests from requesters, comprising: means for
18 receiving at least one request from a requester; means for parsing the at least one request to
19 extract the request specification; means for configuring at least one workflow to deliver the
20 requested reference data based on entitlements of the requester, as well as the selection
21 criteria, sourcing preferences and other preferences contained in requester's request; and
22 means for executing the workflow and delivering the requested reference data to the
23 requester.

24
25 The invention may be used with a multi-source multi-tenant reference data utility, as
26 described herein, for delivering high quality reference data in response to requests from
27 clients, implemented using a shared infrastructure, and also providing added value services
28 using the client's reference data. Data cleansing and quality assurance of the received data
29 with full tracking of the sourcing of each value, storage of resulting entity values in a
30 repository which allows retrievals and enforces source based entitlements, and delivery of
31 retrieved data in the form of on demand datasets supporting a wide range of client application
32 needs, may be utilized. An advantageous implementation has additional services for reporting
33 on data quality and usage, a selection of value adding data driven computations and business

document storage. As noted above, by using a shared infrastructure and amortizing the costs of data quality assurance across a plurality of clients, while ensuring that clients only receive values from data sources to which they are licensed, better quality data is delivered at lower cost than other methods currently available.

BRIEF DESCRIPTION OF THE DRAWINGS

These, and further, aspects, advantages, and features of the invention will be more apparent from the following detailed description of an advantageous embodiment and the appended drawings wherein:

Fig. 1A shows an example component structure of the utility.

Fig. 1B shows example contents of a reference data utility repository.

Fig. 2 shows an example of a top level flow of request processing by the utility.

Fig. 3A shows an example flowchart of processing an arriving source dataset.

Fig. 3B shows an example flowchart of processing client delivery requests.

Fig. 3C shows an example flowchart of processing source, client and entitlement metadata.

Fig. 3D shows an example flowchart of processing value added service requests.

Fig. 3E shows an example flowchart of processing reporting and central service requests.

Fig. 4A shows an example flowchart of processing a data based computation service request.

Fig. 4B shows an example flowchart of processing a business document store or access request.

Fig. 4C shows an example flowchart of processing a business document validation request.

Fig. 4D shows an example flowchart of processing a reference data choreography request.

Fig. 5A shows example types of report from the utility.

Fig. 5B shows example types of utility management service.

Fig. 6 shows scalability, availability and geographic dispersion properties of the utility.

Fig. 7A is an example of a flowchart for managing information and associated source based entitlements in a multi-source multi-tenant data repository.

Fig. 7B is an example of a flow chart for interleaved handling of arriving information, source based entitlements and retrieval requests at the multi-source multi-tenant data repository.

Fig. 8A is an example of an organization of a repository.

Fig. 8B is an example of an organization of an entity in the repository.

Fig. 8C is an example of an organization of item instance within an entity.

Fig. 8D is an example of an organization of a versioned attribute in an item instance.

Fig. 9 is an example of a flowchart for inserting information elements with sourcing annotations into the repository.

1 Fig. 10 is an example of a flowchart for maintaining source-based entitlement information.
2 Fig. 11A is an example of a flowchart for responding to requests to return information
3 elements from the repository based on requester preferences.
4 Fig. 11B is an example of a flowchart interpreting a retrieval request.
5 Fig. 11C is an example of a flowchart for getting the item and item information selection
6 predicates.
7 Fig. 11D is an example of a flowchart for locating requested information elements.
8 Fig. 11E is an example flowchart for enforcing entitlements by filtering retrieved values
9 Fig. 12A shows an overview of the data acquisition and quality enhancement component.
10 Fig. 12B shows an overview of cross-source cleansing.
11 Fig. 13 shows a flowchart of validation, normalization, single-source cleansing and cross-
12 source processing.
13 Fig. 14 shows a flowchart of validation of a single-source dataset.
14 Fig. 15 shows a flowchart of normalization of a source input stream.
15 Fig. 16 shows a flowchart of cleansing of a source input stream.
16 Fig. 17 shows a flowchart of correcting validation errors.
17 Fig. 18A shows a flowchart of correcting normalization errors.
18 Fig. 18B shows a flowchart of correcting cleansing errors.
19 Fig. 19 shows a flowchart of cross-source processing.
20 Figure 20A is a flowchart illustrating producing an on demand dataset in response to an on
21 demand dataset request.
22 Figure 20B is a flowchart illustrating steps in the parsing and analysis of an on demand
23 dataset request specification.
24 Figure 21A is a flowchart illustrating steps in setup of a customized on demand dataset
25 production process.
26 Figure 21B is a flowchart illustrating contents of the library of basic activity building blocks.
27 Figure 22A is a flowchart illustrating structure of an on demand dataset request specification.
28 Figure 22B is a flowchart illustrating an on demand mode case tree.
29 Figure 23A is a flowchart illustrating processing steps in an on demand dataset production
30 process.
31 Figure 23B is a flowchart for retrieve values and insert into delivery dataset step.
32 Figure 23C is a flowchart for an execute delivery instance step.
33

1 DEFINITIONS

2 Attribute – An attribute consists of an attribute name and an attribute value. Example:
3 attribute name = “Exchange where traded”; and attribute value = “NYSE”. Each attribute
4 value in an attribute has a single evolutionary history leading to its creation and has at least
5 one source. Within the repository, multiple versions of the same attribute form versioned
6 attributes. In an advantageous embodiment, sourcing and event information about each
7 attribute is stored in the ETSDT of the versioned attribute.

8
9 Attribute selection – A list of attributes or a predicate on attribute values, identifying the
10 particular attribute values of the selected repository entity to be returned as the output of the
11 request.

12
13 Business document storage service – A service to store business documents in the reference
14 data utility and provide access to them to the owning or to other entitled clients. Each
15 business document may have associated with it validation and data choreography functions
16 which provide added value to clients using the stored business document in their business
17 operations. These added value capabilities can make use of the requesting client’s entitled
18 reference data.

19
20 Client – A customer of the reference data utility. Each client is associated with tenant of the
21 multi-source multi tenant repository in which data is stored on behalf of multiple clients. A
22 tenant may have one or more clients, each client has a subset of the entitlements of the tenant.
23 Administration of client entitlements it typically left to the tenant, but may be offered as a
24 service by the utility. At any point in time there can be multiple agents or programs acting on
25 behalf of a client and making requests on the reference data utility. Each of these agents is
26 then perceived by the reference utility or by components of the reference data utility as a
27 requester. Requests on behalf of a client are for either the delivery of data, or for the
28 execution of added value services, or for the provision of centralized services such as
29 reporting or customer service. Each client is made visible to the reference data utility via a
30 meta data request defining its properties, authorizations, contract protocols, service level and
31 contract agreements, and data and service entitlements. This information is summarized in
32 the client profile.

1 Client profile – A set of information characterizing the allowed behaviors and preferences of
2 a reference data utility client. This will typically include information characterizing the
3 identity, authentication procedures, contact protocols, authorizations and authorization update
4 procedure, Service level agreements, billing arrangements, reporting processes, and
5 entitlement update procedures for that client. The set of client profiles is used by the
6 reference data utility to administer and configure data and associated service deliveries for its
7 collection of clients.

8
9 Data cleansing – The process of determining for each source dataset whether the arriving
10 items conform to that source dataset's source specification and validating the completeness
11 and correctness of attributes received in each item. Data cleansing comprises: acquisition,
12 item validation, item normalization, source dataset specific item cleansing, and multi-source
13 item instance comparison and value selection.

14
15 Data driven computational service – A function or business computation stored in the
16 reference data utility which can be invoked on request from a client of the utility. It is an
17 example of a value-add service which can be provided with a reference data utility. Each
18 data driven computational service has a unique provider who made this service available in
19 the reference data utility. The provider grants entitlements to use the service to some set of
20 clients of the utility. Data driven computational service definitions include data input and
21 output definitions characterizing the reference data they need as input and return as results
22 from each service instance. Instances (invocations) of the data driven computational service
23 execute the service by applying a computation to a particular set of input data provided by the
24 requester and returning a set of output data which becomes the property of the requester and
25 is either delivered to them or stored for them in the repository. On demand data sets are used
26 to insulate the function provider from the specific input and output data transfer and format
27 requirements of each requester. Example: computing a valuation function on a portfolio of
28 complex instruments.

29
30 Data driven computational service registry – A directory with descriptions, and access
31 information for all of the data driven computational services which have been made available
32 at this Reference Data Utility by providers. This registry of value-add services has associated
33 entitlement management enforced by the standard entitlement management facilities of the

reference data utility so that the provider of a data driven computational service can grant entitlement to execute it to specific clients of the reference data utility. Appropriate SLA, billing and reporting arrangements will be put in place when this is done.

Data driven computational service provider – Any party which has made available at least one data driven computational service in a reference data utility for use by clients of the utility. The provider could itself be a client of the utility making this computational service available to others; it could be an agent of the utility making it available as an added value service to some client or it could be an entirely independent third party. The provider of an added value computational service controls entitlement to it.

Data evolution event – Any event resulting in a change to an information element or source element, including deletion and creation of information elements or source elements. Each event includes, at a minimum, an identifier, a timestamp, at least one source of the event, as well as any agents of the event and sufficient information to correlate the event with the information element or source element to which it pertains. Extended attributes of the data evolution event include various additional identifiers, textual descriptions, classifications, etc. The shorter "event" is also used for the same concept.

Delivery dataset – A block of data delivered at one time to the requestor as part of delivery of an on-demand data set. A delivery dataset may be a large or small amount of data.

Delivery instance – The act of transferring a delivery dataset at a point in time to a requestor as part of delivering an on-demand dataset.

Entitlement – A requester's right to access and receive information provided by sources and item instance processes. If a particular attribute value was provided by Source X, but appears in an item instance maintained by item instance process P, then a requester is entitled to this item instance attribute value only if entitled both to source X and item instance process P.

Entitlement repository – An information repository which maintains a listing of: all identified requesters, all sources, all item instance processes, and the entitlement of each identified requester to each source and item instance process.

1 Entity selection – A list of repository entities or a predicate on attributes of repository
2 entities, determining the set of entities for which the request is to return information.

3 Evolutionarily tracked source data tag (ETSDT) – A collection of information reflecting all
4 events in the history of an entity, item instance or versioned attribute. The ETSDT records
5 version as well as all sources and agents of such events. In an advantageous embodiment,
6 ETSDT's are attached to: each repository entity, each item instance, and each versioned
7 attribute of each item instance. In alternate embodiments, ETSDTs may be grouped, split or
8 attached to alternative information elements.

9
10 Information element – One of: a repository entity, an item instance, a versioned attribute, an
11 attribute or a property.

12
13 Item instance – Information on all attributes of a repository entity provided from a single
14 source or item instance process. An item instance comprises a collection of versioned
15 attributes. Item instances carry source information identifying the source or item instance
16 process used to create them. Example: description of IBM stock generated by a comparison
17 and selection process based on information from Vendor A, Vendor B, Vendor C. Some item
18 instances are single source, e.g. data from Vendor A on a particular IBM bond. Other item
19 instances are multi-source and created by an item instance process, e.g. data on a particular
20 IBM bond generated by running a comparison process on a set of sources. Entitlements need
21 to be able to grant access both to individual sources and to item instance processes and their
22 generated item instances. Attributes arriving from the same source at different times may
23 lead to: those being considered separate source datasets leading to creation of separate item
24 instances for each such source dataset, and those being considered timed arrivals within the
25 same source dataset hence included as versioned values within a single item instance.

26
27 Item instance process – A process used to review, validate, cleanse, filter or select from a
28 dataset, or multiple datasets, yielding item instances; also any processes used to review,
29 validate, cleanse, filter or otherwise affect existing item instances. Item instance processes
30 can reflect a single source process (also referred to as “source-specific” elsewhere in this
31 document), as well as processes that utilize data from multiple sources. Composite item
32 instance processes are also possible; “normalized” and “normalized, single source cleansed”
33 are examples of a simple and composite item instance processes, respectively.

1 Metadata – Descriptive information about an information element. Examples: Internal
2 identifiers, timestamps, classification information, textual descriptions.

3
4 Multi-source multi-tenant data repository – A repository with a plurality of entitlement-
5 granting sources and a plurality of tenants that independently arrange receipt of said
6 entitlements with both sources and the repository owner.

7
8 Normalization – For each source item in a source dataset, determining the referred entity
9 about which that item contains information and converting the attributes in the item to be
10 compatible with the target description for the repository entity corresponding to that referred
11 entity. This may include changing the attribute value to a target form.

12
13 On-demand dataset – A logical stream of data created and delivered dynamically via a
14 generated customized run-time process in response to an on-demand dataset request. The data
15 in the on-demand dataset comes from information retrieved from a multi-source multi-tenant
16 data repository. The on-demand dataset is delivered as either a single delivery instance or as
17 a sequence of delivery instances.

18
19 On demand dataset request – A request to create and deliver an on-demand dataset. The
20 description of the requested data is passed as part of the request.

21
22 On demand dataset request specification – The part of an on-demand dataset request that
23 describes the requested data. It describes the contents, sourcing policy, format and delivery
24 specifics of the on-demand dataset.

25
26 On demand source – A source of data from which data can be pulled into the reference data
27 utility, usually with input processing, cleansing and quality assurance as it is received, in
28 response to a request for that data from a client of the utility. Once imported into the utility
29 and stored in the utility's multi-source multi tenant repository, the data can be delivered to
30 other entitled clients.

31
32 Property – Information that does not require versioning because it is public or otherwise
33 generally available for distribution to all tenants of the repository (such as metadata).

1 Information contained within properties can typically be used to make generic requests
2 against the repository at a level which does not require checking entitlements. A property can
3 apply to a repository entity or an item instance. Example: In response to the inquiry; "How
4 many stocks exist in the repository," stock is a piece of classification information required.
5 Because it is inherently publicly available data, it can be exposed as a property, rather than a
6 versioned attribute.

7
8 Reference Data Utility – A common shared infrastructure used to provide cleansed and
9 enhanced reference information from multiple sources as a service to a collection of clients. It
10 may also provide value-add services and general utility support services along with delivery
11 of reference data. The common shared infrastructure includes a multi-source, multi-tenant
12 repository in which raw and enhanced data is stored; it includes shared input processing data
13 cleansing and enhancement in which the source of all information is tracked; it includes on
14 demand dataset delivery allowing entitled data to be selected, retrieved and delivered to all
15 clients matching their delivery specifications; it includes the provision of value added and
16 centralized services. Clients of the reference data repository are tenants of the multi-source,
17 multi-tenant repository component used to store data for the reference data utility. The term
18 reference data utility is often shortened to utility.

19
20 Referred entity – A real world entity described by information stored in the repository.
21 Example: an actual bond issued by IBM, a corporation, a counter party or stock trade.

22
23 Repository – A collection of information consisting of: repository entities, value add services
24 and business documents, in which knowledge of the contributing source and evolutionary
25 history of each piece of information in the collection is maintained.

26 Repository entity – A collection of information stored in the repository describing a single
27 referred entity. A repository entity consists of a set of attributes defining the entity (its
28 metadata, e.g. name, properties) and a collection of item instances each containing additional
29 information on the repository entity added into the repository from an identified source or
30 item instance process. Example: information in the repository characterizing a particular
31 bond issued by IBM, corporation, counter party or stock trade.

1 Repository owner – An organization or corporate entity that owns a repository and makes the
2 repository data services available to tenants subject to their entitlement agreements with
3 sources and additional entitlements to item instance processes of the repository.

4
5 Repository access request – A request for access to information stored in the repository from
6 an identified requester. Information required in processing a repository access request
7 includes requester identification, sourcing preference and selection predicate. May also
8 include entity and attribute selections.

9
10 Request specification – Information required in processing a request for information from a
11 multi-source multi-tenant repository. At a minimum, includes requester identification,
12 sourcing preference and selection predicate. May also include entity and attribute selections.

13
14 Requester – An agent making a repository access or other request. This agent may be acting
15 on behalf of a client of the repository or may be acting for the repository, or a computer
16 program acting on behalf of one of these parties. The requester responsible for a request
17 needs to be identified so that entitlements can be enforced in responding to the request.
18 Requesters are uniquely identified by a requester identifier.

19
20 Selection predicate – Specification of those information elements a requester is interested in
21 receiving in response to a request for information from a multi-source multi-tenant
22 repository. A component of the request specification, it most often refers to repository
23 entities, item instances and versioned attributes.

24 Source – An identifiable supplier of one or more source datasets each containing information
25 on referred entities. A source may be uniquely identified by its source identifier. Example:
26 Vendor A and Vendor C.

27
28 Source accuracy – The frequency with which a source-supplied attribute value coincides with
29 the selected value (recommended value) resulting from some multi-source item instance
30 process. This provides an objective measure of the relative quality of different sources of
31 information to the repository.

1 Source attribute – Source attributes make up source items in source datasets. See source item
2 definition below. For example, if a source item represents common stock of company X as
3 received from some source, the exchange on which the stock of company X trades is a source
4 attribute. Source attributes are normally represented as name-value pairs.

6 Source dataset – A collection of source items from a specific identified source; source
7 datasets may become available at a specific point in time, may become available continuously
8 or may be fetched on-demand by a sequence for requests. Example: Vendor A Public Bond
9 Information Service. Source datasets are uniquely identified by a source dataset identifier.
10 The source identifier for the providing source may or may not be part of the source dataset
11 identifier.

13 Source dataset description – Information describing the structure, content of the source
14 dataset and any constraints on values of attributes appearing in items of the source dataset.
15 The source description is provided by the source responsible for the source dataset.

17 Source dataset identifier – See the definition of source dataset above.

19 Source element – a source item or a source attribute.

21 Source identifier – See the definition of source above.

23 Source item – Information contained in a single source dataset that describes a particular
24 referred entity. A source item is a collection of source attributes that may include any or all of
25 the attributes of the referred entity.

27 Source usage – The source usage by a client of a particular source is the number of times that
28 a request from that client results in delivery of information provided by that source. This
29 may be provided as the total usage from each source within some fixed period of time. Note
30 that usage of a source may be explicit or implicit; explicit usage is when this source was
31 selected through a specific requester policy identifying the source; implicit usage is when the
32 preference is for some multi-source item instance and the source was a supplier of the
33 selected value for that item instance.

1 Source profile – A source profile contains information characterizing the behavior of a data
2 source used by a reference data utility. This will typically include information on the
3 identity, authentication procedures, contact information, authorizations, input formats, source
4 data delivery protocols, data correction protocols, entitlement updates and reporting
5 arrangements for that data source. The reference data utility uses its collection of source
6 profiles to administer and configure input processing and cleansing of data received from all
7 data sources.

8
9 Sourcing, sourcing information – A source of data; can be an item instance process (e.g.
10 cross-source comparison and selection process) or a specific data provider (e.g. Vendor A).

11
12 Sourcing preference – An ordered list of sources and item instance processes; the requester
13 would prefer that attributes and attributes returned as output from the request come from item
14 instances early in this order. Since the processing of requests by the repository enforces
15 entitlement, a requester will not always receive attributes and values from the first choice
16 source in this list but has partial control of the values selected for return.

17
18 Target dataset – Information describing the structure, contents and constraints on repository
19 entity information, including item instances, versioned attributes and attributes as stored in
20 the repository. Note that this is a target description from the perspective of input cleansing
21 only. The clients of the repository may regard the target description as the schema for the
22 repository entities which from their perspective is the provider of their reference information.

23
24 Tenant – An organization, individual or corporate entity which arranges to be a user of a
25 reference data utility or more specifically of a repository and may arrange with the utility or
26 repository owner and sources to be entitled to information and services. Tenants may pass on
27 entitlements to identified clients acting on their behalf.

28
29 Topic – A repository entity property used for hierarchical organization within the repository.
30 For further granularity, topics may be divided into subtopics. In principle, every repository
31 entity in the data repository is uniquely located in this hierarchical topic space. Example:
32 Financial instrument definitions or corporate ownership hierarchies are examples of topics in
33 a financial reference data repository. The financial instrument definition topic may be

1 decomposed into subtopics such as common stock definitions and bond definitions; within
2 bond definitions further divided into corporate bonds and government backed bonds, and so
3 on.

4
5 Value added service – In the context of a reference data utility, an optional service providing
6 added value to clients of the reference data utility which is indirectly related to reference data
7 and takes advantage of capabilities of the base reference data utility. Data driven
8 computational services and business document services are examples of value added services
9 optionally provided with a reference data utility. Clients obtain a value added service by
10 issuing a value added service request to the reference data utility. Examples of value added
11 services usefully provided with a reference data utility include data driven computational
12 services and business document storage services.

13
14 Value added service request – A request to the reference data utility from a client to obtain a
15 value added service.

16
17 Versioned attribute – A collection of one or more versions of the same attribute, wherein each
18 version was produced by a different source or sources. In an advantageous embodiment, an
19 attribute name and a collection of one or more attribute values. An advantageous
20 embodiment for organizing and storing a versioned attribute in the repository is as a
21 collection of attributes (as defined above) where all attributes in the collection have the same
22 attribute name. This organization allows a versioned attribute to be constructed in the
23 repository by moving or copying attributes from a source dataset into a versioned attribute in
24 an item instance, as well as by adding additional attributes as modified attribute values are
25 created by some value enhancement process. A versioned attribute has an ETSDT in which
26 all events and sources pertaining to attribute values in the versioned attribute are recorded.
27 Hence, multiple “values” (multiple contained attributes in an advantageous embodiment) can
28 exist within a single versioned attribute in an item instance, pertaining either to a value from
29 the same original source that was modified by some item instance process(es), or to a value
30 that was composed or selected from multiple original sources.

DETAILED DESCRIPTION OF THE INVENTION

General Organization

The invention will be described in four sections each addressing a separate aspect, as also used above in the Summary of the Invention. The first section describes the method and operation of a reference data utility with properties that it is outsourceable, shareable, able to support multiple tenants and multiple sources of data and to enforce entitlement and privacy rights to its contained information. Each source may grant entitlements to information derived from its data to any combination of tenants. The information entitled to each tenant depends on the sources used to derive it and the enhancement processes applied to the source data. The section also describes optional additional document choreography and computational services which can be provided by the reference data utility to increase its value to tenants. In an advantageous embodiment a reference data utility includes such value add services.

The second section describes the structure and methods for forming and operating a repository in which information is stored, access to the stored information is granted to requesters and entitlement rights relating to the source and enhancement processing of the data are enforced by tagging individual data elements with a summary of the history by which they were generated.

In an advantageous embodiment, a reference data utility uses such a repository as an information storage and access method for its reference data.

The third section describes a method and organization for performing scalable data cleansing and enhancement of arriving reference information in which both single data source enhancement processing and multiple data source comparison and enhancement processing are supported while the method still maintains full knowledge of all sources used in deriving reference data elements. In an advantageous embodiment, a reference data utility applies this data cleansing and enhancement processing to arriving information from sources as its input method.

1 The fourth and final section describes a method and organization for scalable on demand
2 delivery of reference data from a repository to requesting clients in which a wide variety of
3 client needs for different delivery content, format and mode of data delivery are
4 accommodated. In an advantageous embodiment, a reference data utility uses this method
5 to deliver data from the utility to clients associated with tenants of the utility in a scalable
6 manner as its output method.

7
8 A. General Structure and Method of Operation of the Reference Data Utility

9 The invention, in a first major aspect, is a method and novel system organization for forming
10 and maintaining a multi-source multi-tenant reference data utility delivering high quality
11 reference data in response to requests from clients, implemented using a shared infrastructure,
12 and also providing added value services using the client's reference data. An advantageous
13 implementation offers additional services for reporting data quality and usage, a selection of
14 value added data driven computations and business document storage.

15
16 The method is effectively an "assembly line approach" to data gathering, quality assurance,
17 storage and delivery of reference data. The ability to support a wide range of client
18 requirements for different topics, sources, qualities, modes and formats, organized as an
19 automated extensible system, provides a valuable service by enabling the expensive but
20 critical human expertise and review functions to be centralized and highly leveraged. The
21 design of the utility allows for the efficient global sourcing of data, affording significant
22 economies of scale. The component structure allows for the efficient global distribution of
23 different functions of the utility; this also enables the ability to substitute components and
24 respond to change as business develops. Clients of the utility receive their reference data
25 from one or more sources indirectly through the utility which gives them the flexibility to
26 reconfiguring their applications to receive reference data from different sources. Gathering
27 and providing uniform quality assurance of reference data on a broad range of topics in a
28 single utility service increases the likelihood that individual client applications of clients will
29 discover and use the best available reference data values. The maintenance and enforcement
30 of source based entitlements in a multi-source multi-tenant shared repository allows a single
31 shared infrastructure to accommodate multiple tenant organizations, with independent
32 departments and applications both across and within tenant organizations to make their own
33 arrangements to license data from supported sources. The reference data utility assures the

1 data sources, through audit log support, that each client of the utility is receiving values
2 derived only from sources to which they are licensed. This auditable assurance is based on
3 the method providing full transparency of the data for each repository entity value. Full
4 sourcing documentation is available; each delivery of a value to a client is logged, identifying
5 the available value and the user access. Regulatory compliance in handling reference data is
6 an expensive proposition for each individual financial services business; using the reference
7 data utility repository to provide this via a uniform mechanism whose cost is amortized
8 across all client organizations offers cost advantages. A standard reference data source
9 promotes coherence and consistency within the industry.

10
11 Delivering reference data through a shared repository, with tracked data sources and access,
12 creates a marketplace in which higher level financial service providers can offer their models
13 to many clients and be assured of receiving reliable usage information for contract
14 enforcement or billing. Clients use these higher level services on data in the repository to
15 which they are entitled, with the assurance that data access rules will be enforced and
16 monitored to assure compliance with data access and transfer regulations

17
18 The reference data utility provides monitoring, reporting and customer service as expected in
19 a utility solution. A valuable point of novelty is that the utility provides an objective measure
20 of the accuracy and quality of different available data sources based on its processes for
21 comparing values for the same attribute from different sources.

22
23 The above capabilities are provided in an environment in which the security and privacy of
24 client actions is maintained. No client or data vendor is able to discover information about
25 another's data, queries or other actions taken by the repository to support them.

26
27 The reference data utility provides benefit through a centralized governance scheme for
28 access to operations and data within the utility, allowing clients and data vendors appropriate
29 access to update and self manage resources in the utility which are either invisible or
30 appropriately reflected to other actors.

31
32 The method is described herein as it applies to reference data used by Financial Services
33 businesses. This method for provisioning a multi-source multi-tenant data repository

1 providing shared access to data used for reference by an organization has many other possible
2 areas of application. Access to consumer credit information, government regulation and
3 registration information, and telecommunications usage information are three additional
4 examples where the method would be useful. Characteristics of contexts where the method
5 will be useful and of reference data are: (1) the information comes from many sources (2)
6 there are multiple users potentially in independent organizations needing access to the same
7 information but potentially with different source entitlement rights (3) the referenced
8 information is accessed by users largely in read-only mode except when they participate in
9 correcting invalid values (4) high quality timely information is both valuable and complex to
10 gather hence the efficiencies from a utility approach, shared infrastructure and shared data
11 quality enhancement provide significant benefit (5) entitlement enforcement and privacy
12 management is provided by the repository. Although the invention is described herein in the
13 context of financial services reference data which is one important area of application, the
14 approach disclosed herein, enabling an effective repository to provide data access meeting the
15 requirements above, will have value in any context with these requirements.

16
17 Fig 1A provides an overview of the major functional units and component structure of the
18 reference data utility and its associated operational environment. In Fig. 1A, polygon 1,
19 delineates the boundaries of the reference data utility. Circles representing clients 6, 7, 8 and
20 9, of the utility 1, appear on the right. Dashed boxes 2, 3, 4, and 5, representing different
21 types of data and service sources, appear on the left. Reference data utility 1 can have
22 multiple sources supplying data and other inputs. For illustration purposes Fig. 1A uses
23 seven data sources S1, S2, S3, S4, S5, S7 and S8. These data sources are classified into three
24 types as described below. The number of sources of each type is not limited.

25
26 Source S1, source S2 and source S3, shown as ellipses 10, 11, 12 respectively, in box2 of
27 Fig.1A. represent licensed pre-qualified data sources. The data received from these sources is
28 proprietary. Each source may independently license delivery of its data to clients of the
29 reference data utility 1. As the reference data utility 1 enhances, stores and delivers data
30 derived from these sources, it maintains knowledge of the source of each received data item
31 and of any values derived from it. Furthermore the reference data utility 1 enforces
32 entitlements ensuring that each client receives data only from sources to which it is entitled.

Source S4 and source S5, represented by ellipses 13 and 14, in box 3, are in the unlicensed and public category of raw source data that is continually used and monitored by the reference data utility 1. Because this data is public and unlicensed, no incremental payment for distribution of the values is expected. This information is typically incorporated into the repository 20 (discussed below) of reference data utility 1, as properties of repository entities rather than entity attributes which are explicitly versioned and tracked. Data in this category can be used freely by the reference data utility 1 to validate or augment other streams of data and values. Source information in this category includes news reports of corporate actions and published registries of financial instrument names and properties. While data in this category does not require tracking in order to enforce entitlements, operators of the utility 1 may also choose to track this type of data for various reasons such as providing auditable sourcing information so that the quality of public sources can be analyzed over time to eliminate public sources of poor quality data.

Source S7 and source S8, represented by ellipses 15 and 16, in box 4, are in the category of on demand data sources providing data that is only fetched on demand as a result of a request from a utility client. Thus, it is distinguished from pushed streams of data received from regular licensed data vendors and from the continuously monitored public data which affects the interpretation of intensively used data in box 3. The definition and pricing information on infrequently traded instruments, such as a bond issued by a local authority or public service organization, is an example of information in the category represented by box 4. When a specific reference data utility client (most often as part of a retail banking operation) requires this information, an action by the repository will request values for that reference item from appropriate sources and perform standard data validation, storage and delivery processing.

Service V1 and service V2, represented by ellipses 17 and 18, in box 5, are a different category of non-data sources providing input to the utility 1. Data driven computational services are made available to the utility 1 by third party providers and are used to add value to clients' data. The reference data utility 1 provides a marketplace to help clients find relevant value added services and manages the execution of data driven computational services on clients' data. A client of the utility can only use entitled services, and a service, while acting on behalf of a client, can only access data to which the client is entitled. As part of this processing, each client use of a service is monitored and recorded by the utility 1.

1 Using this information, the reference data utility 1 can efficiently charge and collect from
2 clients for their data driven computational service usage on behalf of and in conjunction with
3 the service provider. In an alternative embodiment, the utility meters the use of computation
4 services by clients and invoicing and payment are handled by the provider of the service. The
5 utility can mix these two implementations, billing for some computational services and not
6 for others. Higher level value added services are optional. The utility 1 enables their
7 existence. The functions they add to the utility 1 provide significant incremental value for the
8 utility's clients.

9
10 Each client 6, 7, 8 and 9 may be an independent enterprise or a department within an
11 enterprise. Each client receives high quality data values from the utility 1 in the form of
12 delivered on demand datasets. Each on demand dataset is either a response to standing
13 subscriptions (representing a sustained interest in regular or quasi real time updates on
14 particular reference item values) or a response to a one-time ad hoc query. Each client will
15 also control how, when, and in what form data values are delivered. In order for the utility to
16 be widely attractive, it is important that wide ranging and flexible data delivery services be
17 defined so that each customer can have data values delivered to them in a convenient format
18 without customized engineering work inside the utility 1. Flexible delivery with customized
19 support embedded into the system structure of utility 1 enables amortization of data costs
20 across many tenants, hence realizing the multi-source multi-tenant data utility 1 as an
21 advantageous system and method.

22
23 Boxes 19, 20 and 21 represent the three primary components involved in the flow of data
24 values through the system; from raw data sources through delivery to customers of utility 1.
25 Box 19 represents the data acquisition and quality assurance component responsible for
26 gathering data values into the repository system and assuring the high quality of the data. Box
27 20 represents the reference data utility repository component responsible for storage and
28 access management of all persistent information needed in the repository. Box 21 represents
29 the delivery component responsible for capturing the on demand dataset request
30 specifications of each requester and constructing the automated delivery procedure to deliver
31 that information.

1 Inside box 19, the data acquisition and quality enhancement components or boxes 22, 23 and
2 24, represent the independent input and quality processing for separate data topics T1, T2 and
3 T3, respectively. Each topic can have an arbitrary number of sources providing data for it; a
4 single topic can combine data from any combination of licensed pre-qualified data sources,
5 free access data sources and qualified on demand sources. For example, box 24 indicates that
6 free source S5, ellipse 14, and on demand sources S7, ellipse 15, and S8, ellipse 16, are all
7 supplying data on topic T3. Box 23 is receiving data from pre qualified source S3, ellipse 12,
8 and free source S4, ellipse 13. Box 22 receives data on topic T1 from pre-qualified sources
9 S1, ellipse 10, source S2, ellipse 11 and source S3, ellipse 12. Arrow 39 shows the data
10 received or generated during data acquisition and quality assurance being stored in the
11 repository 20. In order for the reference data utility to enforce source based entitlements to
12 data for its multiple clients, knowledge of all sources contributing to each data value must be
13 maintained through the processing of box 19. The data acquisition and quality enhancement
14 processing of box 19 also supports both single source values, based on analysis of one
15 licensed data source's data describing a referred entity, and multi-source values, obtained by
16 comparing values from multiple sources describing a single referred entity attribute, and
17 selecting a preferred or recommended value from the set.

18
19 A method for enabling scalable cleansing and value enhancement of reference data by
20 employing evolutionarily tracked source data tags meeting the above needs is described
21 below.

22
23 Generated data to which data acquisition and enhancement processing is applied in box 19
24 can also arrive as the output of a data driven computational service or as data retrieved from
25 an on demand data source in response to some client request. The types of data that can be
26 stored in the repository are described in Fig. 1B.

27
28 Box 21 is the client delivery component; boxes 30, 31, 32 and 33 represent the on demand
29 dataset processing for each client. Specifically, box 30 is the delivery processing for client
30 C1, circle 6, box 31 is the delivery processing for client C2, circle 7, box 32 is the delivery
31 processing for client C3, circle 8, and box 33 is the delivery processing for client C4, circle 9.
32 The reference data utility 1 can have an arbitrary number of clients, concurrently or serially.
33 For illustration purposes four clients C1, C2, C3, C4 are used. For each client, independent

1 processing in response to requests from that client selects values of entities of interest and
2 delivers them via appropriate delivery protocols and transforms. Arrow 41 represents
3 retrieval requests generated as part of on demand dataset processing being presented to the
4 repository 20 of reference data utility 1 and the resulting return of information from where it
5 is stored in the repository 20 of reference data utility 1 for delivery to a client. Thus, arrow
6 41 shows that repository 20 provides requested reference data values as needed by the client
7 data delivery component (box 21).

8
9 Other types of functions are included within the context of the utility. Box 34 represents
10 utility management and report generation services. The report generation service creates one
11 time or periodic reports for clients and data sources. These reports provide information on
12 utilization, delivery summaries, accuracy and similar aspects of service level reporting. Box
13 35 represents the general client service function which assists clients with operational
14 requests, problem diagnosis, customer questions, concerns or proposed corrections for
15 specific reference values, etc.

16
17 Box 36 represents additional value added services offered by the utility 1. This includes data
18 mart hosting and data transform services, data driven computational services applied on
19 request to the clients' data by the utility 1, and business document storage services.

20
21 Ellipse 37 represents the pool of human topic experts who provide key decision making for
22 manual processes within the utility 1. The expertise of these people is also likely to be needed
23 to participate in client service functions.

24
25 Arrow 39 shows data from the data acquisition and quality enhancement component (box 19)
26 flowing into the repository 20.

27
28 Arrow 40 shows that the instances of value add services use reference data entitled to the
29 invoking client while they are running. Arrow 38 shows that the repository 20 will canvas on
30 demand data sources to gather additional information. Arrow 42 shows an example of client
31 invoking the value added services (box 36), reporting and utility management (box 34), and
32 general services (box 35) of the reference data utility 1.

Fig. 1B shows an example of information stored in a reference data utility repository. This information includes entitlement managed entity data in box 50. Entitlement managed entity data includes entity data derived from a single source, box 26, and entity values derived from comparisons of multiple sources providing alternate values from which a preferred or recommended value has been selected, box 27. A method for provisioning and maintaining a multi-source multi-tenant data repository with entitlement management based on source tracking of reference data is described below.

Other data elements in Fig.1B show information maintained in the repository 20 of the reference data utility 1 that is not organized as entitlement managed entity data. Entitlements are maintained and enforced on all of this data as appropriate using access control stored in an entitlement repository shown as data element 53. As noted above, entitlement management of entity data is source based and requires maintaining information on all data sources which have contributed to the derivation of each particular value. For other data in the repository, entitlement management consists of simple access control, using techniques known to the art to record for each object, which clients have access to it and which operations are available to them. The preferred embodiment as shown includes an entitlement repository integrated into the repository 20 of reference data utility 1; an alternate embodiment maintains equivalent information in an independent entitlement repository.

The non-entity data structures stored in the reference data repository with access control provided through the entitlement repository are listed next. Data element 25 represents logs of data as received from the data sources. These logs are maintained for non-repudiation and information source tracing. Data element 29 represents logs of data delivered to clients of the utility 1, recording exactly what values were delivered at what times to each client. The client delivery logs are maintained for audit, transparency, regulation compliance and billing purposes. Data element 28 represents the normalization tables and metadata used to combine input from independent sources and to determine when information from multiple sources is describing a single referred entity. Rules associated with cleansing, normalization, and validation used in the processing of Fig. 1A, box 19, can also be stored in the repository 20 of reference data utility 1. Data element 51 represents source profiles. Each source profile contains information about the interaction protocols, source formatting and encoding used by a data or other input source. Data element 52 represents client profiles. Each client profile

contains tenant information, contact information, billing and reporting requirements, operational authorizations, sourcing, format and delivery policy preferences for a client of the reference data utility. Tennant profiles are a special form of client profile which characterize the overall entitlements that each client of the tenant has. Source and client profiles are used in the configuration operations of the reference data utility 1 to ensure flexible, independent adaptation to changes in source and client characteristics and to the introduction of new sources and clients.

Data elements 54, 55, 56, 57, 58, 59, 60, 61, and 62 are optional elements used to support reporting and added value services associated with clients' reference data. Data elements 54, 55, 56 and 61 are reports accumulated and saved in the repository 20 of reference data utility 1 for data sources, clients' function providers and regulators, respectively. Data element 57 is a registry of added value data driven computational services. Data element 60 represents the data driven computational functions in executable form. Data element 58 represents client data sets produced as on demand datasets or as the output of a data driven computational services. Data element 59 represents the business document repository. Data element 62 management reports generated for the operation of the reference data utility.

Figure 2 provides a top level view of the processing of requests by the utility in the form of a flow chart. In this and following flowchart diagrams, solid lines represent control flows and dashed lines represent data movement. Box 100, bounding this diagram, corresponds to the control flow of the overall method of the invention and reference data utility 1 introduced in Fig. 1A and Fig. 1B. Dashed arrow 200 represents all the different requests for reference data utility processing which are handled by this control flow.

Control flows into box 100 from the left into element 201, representing the arrival of a request for processing at the utility 1. A request for processing may originate with data sources, clients of the utility, data driven computational service providers, or staff of the utility itself. Element 201 also includes authentication processing to uniquely identify the person or agent making the processing request, authorization checking to determine that the requester is authorized to make the request and logging the request to ensure that there is an auditable record of all processing done by the utility.

Decision element 202 differentiates the processing of requests by request type, showing a different processing path for each type of request arriving at the utility. The path through outcome element 203 handles new source datasets arriving at the utility. An arriving source dataset is processed in element 208; the description of this processing is elaborated upon with Fig. 3A. The combination of the processing of 203 and 208 is the function performed in block 19 of Fig. 1A. The path through outcome element 204 handles a request from a client for delivery of reference data from the utility. Processing of client delivery requests is handled in element 209; the description of this processing is elaborated upon in Fig. 3B. The combination of block 204 and 209 corresponds to the processing of block 21 in Figure 1A. The path through outcome element 205 handles profile updates and entitlement updates. These requests identify new clients, new sources, new entitlements to data or value-add functions, or changes to previously registered information of these types. Processing of these requests is handled in element 210; the description of this processing is elaborated upon in Fig. 3C. The processing of blocks 205 and 210 is part of handling data within block 20 of Fig. 1A. The path through outcome element 206 handles requests for processing associated with value added services using information in the utility to provided clients with optional additional capabilities. The processing of these requests is handled in box 211 and elaborated upon in Fig. 3D. The processing of blocks 206 and 211 corresponds to the processing of block 36 in Fig. 1A. The path through outcome element 207 handles requests for general services including the generation of reports by the utility; processing of these requests is handled in box 212 and elaborated upon in Fig. 3E. The processing of blocks 207 and 212 is split between block 35 of Fig. 1A for general services and block 34 of Fig. 1A for reports and utility management requests. Alternate embodiments will contain the same functions but may organize them into different blocks.

After separate request processing by the utility for each of the different types of processing requests, the control flows converge on decision element 213. This decision element determines whether processing continues with the next request or terminates. In the case of continued processing, control flows back to element 201, providing a loop structure. Each iteration of the loop from element 201 to element 213 handles one request. In the case of terminated request processing, control flows out of box 100 ending the flow of the method.

1 For expository convenience the control flow of Fig. 2 shows the processing of requests
2 sequentially by the reference data utility. Using transaction processing, database and
3 workflow, or other techniques well known in the art, an alternative embodiment of the utility
4 processes requests from many clients, sources, function providers, and utility staff
5 concurrently.

6

7 Exit from the processing of box 100 may occur to shut down the utility. Return to additional
8 request handling in element 201 provides clients of the reference data utility 1 continuously
9 available access to their reference data and associated utility services.

10 Fig. 3A provides a high level flowchart showing the steps in processing a dataset arriving
11 from a source. It is an elaboration of the processing element 208 first introduced in Fig. 2.
12 Arriving data is cleansed and used to generate new values for insertion into the multi-source
13 multi tenant data repository 20 (herein referred to as "repository"). New values may trigger
14 additional deliveries of data to clients. Events in cleansing the data and generating values
15 stored in the repository 20 may be documented and used to update utility reports on the data
16 sourcing process.

17

18 Element 208, bounding the flow in Fig. 3A, shows this flow is an elaboration of the
19 processing of a new source dataset. Control enters element 208 from the top and flows to
20 element 301 where the arriving source dataset is associated with its source. The repository 20
21 will maintain descriptive and processing control information for each data source which it is
22 using. The information about each data source is saved in a source profile in element 51, the
23 set of source profiles. Information in a source profile includes authentication tokens, which
24 the utility can use to verify that the dataset originated with the expected source, definitions of
25 the exact source data formats, other conventions and protocols used by this data source and
26 contact arrangements for handling error correction process with the source, and requests for
27 additional data from this source.

28

29 Data element 51 is a set of source profiles for sources used by utility 1. The dashed arrow
30 from element 51 to element 301 represents the action of element 301 to select the appropriate
31 source profile for the source providing the new dataset and use information from that source
32 profile to refine subsequent processing of the dataset. In an advantageous embodiment,

1 source profiles are stored in the repository 20 on reference data utility 1 as described in Fig.
2 1B.

3
4 The next step in the flow, element 302, provides cleansing and quality assurance of the
5 information in the new source dataset, and generates enhanced values for repository entities
6 and their properties and documents events in the quality assurance and data enhancement
7 processing. This step requires a method for scalable cleansing and value enhancement of
8 reference data with tracking of enhancement events such as that described below.

9
10 One of the actions of the cleansing and data assurance processing is to generate logs of data
11 received from data sources for non repudiation, source tracing and audit purposes. This action
12 is represented by the dashed arrow connecting element 302 to the received data logs, data
13 element 25. In an advantageous embodiment, received data logs are stored in the repository
14 20 of reference data utility 1 as described in Fig. 1B.

15
16 The next step in the control flow, element 303, stores derived values from element 302 as
17 entitlement managed entity data shown as data element 50. This entity data is annotated with
18 origination information for every stored information element so that source based
19 entitlements can be enforced when the utility delivers information to clients. In an
20 advantageous embodiment, as noted in Fig. 1B the entitlement managed entity data is stored
21 in the repository 20 of reference data utility 1. A method for maintaining a multi-source
22 multi-tenant data repository and processing steps to insert new values into it are described in
23 detail below.

24
25 A dashed arrow connecting element 303 with data element 50, the entitlement managed entity
26 data, shows that the derived values are added to this data element. A second dashed arrow
27 from data element 50 to (processing) element 308 shows updates and insertions to the
28 entitlement managed entity data triggering delivery processing to add the new values into an
29 on demand dataset for subsequent delivery to a client. That trigger is described in the
30 delivery processing flow discussed in Fig. 3B.

31
32 During the processing of step 302, events occur in the evolutionary history of entity values.
33 Examples include: the correction of an incorrect value from a source, subsequent

confirmation of a correction from a source, and selection of recommended values based on comparison of corresponding values from multiple sources. . These cleansing events are captured and carry important information about the quality of data arriving from each source. The following step, element 304, is the processing to analyze captured source data quality information and include it in reports generated by the utility for each source on the quality of datasets they provide. A dashed arrow from element 304 shows this information being passed to data element 54, representing source reports. Ongoing processing in the utility maintains reporting on source data quality. Each source can be given access to the utility reports on its provided datasets.

Fig. 3B provides a high level flowchart showing the steps of processing client delivery requests.

Box 209 is elaborated upon below, to show how, within the full utility context, value added data delivery is provided in response to on demand delivery requests from clients of the utility.

An on demand dataset request (herein referred to as “request”) enters the utility in box 311. The first step is to associate the on demand dataset request with a client of the utility and authenticate it. This is done in a standard manner known to practitioners of the art, using one of a number of known methods to verify credentials contained in the delivery request against client profile information stored in the utility’s repository and represented as data element 52. Information contained in the client profile of the requester is retrieved as illustrated by the arrow representing data flow from data element 52 to box 311.

Once the request has been authenticated and a matching client profile found, the step represented by decision box 312 determines whether additional values are gathered before the process of responding to a request, as described below. Independent parsing of the request is done in this step, which, in alternate embodiments, can be combined with parsing done as part of responding to the request. Additional value gathering includes requesting additional input data from on demand sources and dynamically performing a data driven computational service against existing repository data. In an advantageous embodiment, the resulting new data is passed through a data acquisition and quality enhancement process as described in box

19, introduced in Fig. 1A, and then stored in the repository 20 of reference data utility 1. As such, additional value gathering constitutes a separate service offered by the utility that has its own associated entitlements. Therefore, step 312 examines information from the entitlement repository, element 53, to ensure that the requester is entitled to the additional value gathering service. Queries against the currently available entity data in the repository 20 can be made to access its state relative to the request. Other constraints, such as whether a client's requested delivery timeframe accommodates the additional value gathering can be considered. If additional value gathering is required, the appropriate value gathering process is initiated, at box 313. This may include requesting data from an on demand data source 4. The resulting new entity values are added to the entitlement managed entity data shown by the dashed arrow from box 313 to data element 50. Once additional value gathering is complete, or if no additional value gathering is necessary, the process of responding to the request is initiated as described below (box 314). The process includes retrieving entitled data values from the multi-source multi-tenant data repository 20, the repository of the reference data utility, box 50. As the delivery process culminates with the formation and delivery of the on demand dataset to a requester, updates to the client delivery log, element 29, are generated. Box 314 shows updates being generated and added to the client delivery logs in data element 29. Box 315, which follows in the flow, creates and stores client reports on data source utilizations and received data summaries. The dashed arrow connecting box 315 with data element 55 represents this reporting activity. In an advantageous embodiment client delivery logs and client reports are retained in the reference data utility repository as described in Fig. 1B.

Fig. 3C provides a flowchart showing the steps in processing arriving metadata that characterizes sources of data, tenants, clients of the utility and entitlements of particular clients including, entitlements to data from particular sources and entitlements to value-add services. The utility 1 maintains current metadata on sources, clients and entitlements in order to adapt its configuration, and to control its processing of all other requests. Fig. 3C is an elaboration of box 210 first introduced in Fig. 2, also shown as box 210 bounding the control flow in Fig 3C.

Control enters box 210 from the top and flows into decision element 321 which determines the type of the metadata request. Each metadata request is either new information on a

1 source, represented by outcome element 322, new information on a client, represented by
2 outcome element 324, or new information on an entitlement, represented by outcome element
3 328.

4
5 New metadata information characterizing a source is handled in element 323, by creating or
6 updating a source profile. The utility maintains a source profile, data element 51, for each
7 source providing source datasets. These could be base sources providing raw data or
8 processes, (e. g. item instance processes), which creates additional or enhanced data values
9 from other data. If the arriving metadata describes a new source of data, a source profile is
10 created in step 323. If the arriving metadata is an update for a source previously known to
11 the utility, the profile for that source is updated In step 323. The metadata request can also
12 trigger the deletion in this step of a profile for a source which will no longer be used. The
13 source profile contains control information needed to cleanse, quality enhance and transform
14 data from that source into repository entity fields. This includes authentication tokens to
15 validate a source as the origin of arriving data, formats, encodings and protocols for receiving
16 datasets from the source, contact arrangements for correction interactions, reporting
17 arrangements, data access and updated authorizations granted to agents acting for the source.
18 Metadata characterizing item instance processes used to derive enhanced values is similar to
19 raw source data and is handled in the same step.

20
21 New metadata information characterizing a client or tenant of the utility is handled in element
22 325 by creating or updating that client's or tenant's profile. The utility maintains a client
23 profile, data element 52, for each of its clients. If the arriving metadata describes a new
24 client, a client profile is created in step 325. If the arriving metadata is an update for a client
25 previously known to the utility, the profile for that client is updated in step 325. The metadata
26 request can also trigger the deletion in this step of a profile for a client who will no longer be
27 active. The client profile contains information necessary to handle and control processing of
28 requests from that client for data delivery, value-add services, customer service and reporting.
29 This includes authentication tokens to determine when requests have originated with that
30 client or its agents, authorization information identifying and specifying operational access
31 rights for each agent of the client, service level agreements applicable to responses provided
32 by the utility, pricing and volume arrangements with the client, reporting services to be

1 provided by the utility, preferred data outputs and contact information for interactions with
2 the client.

3
4 After updating a source or client profile, control flows to decision element 326 which tests
5 whether a new source or a new client has been introduced. If this is the case processing flows
6 to step 327 which is an update of the entitlement repository 53 with a reference to the new
7 data source or client. This update will allow source based entitlements granted by the new
8 source or granted to the new client to be added into the entitlement repository 53. If,
9 conversely, the test in decision element 326 shows that the metadata update was to the profile
10 for an existing source or client profile, no change to the entitlement repository 53 is needed at
11 this point.

12
13 If the result of the test in decision element 321 was that the new metadata is an entitlement
14 change, control flows via outcome element 328 into the processing block 329 where the
15 entitlement repository 53 is updated to reflect this entitlement metadata.

16
17 A change in entitlements is either a change in source based entitlements to raw entity data, a
18 change in entitlement to a data enhancement process, or a change in simple entitlements to a
19 value added service or other utility object. A change in source based entitlements takes the
20 form of a new modified or deleted grant, granting access to one or more clients to data from
21 one or more sources or item instance processes. The required processing for this case is to
22 make the appropriate change to the list of entitlement grants in the entitlement repository.
23 Representative flows showing application of updates to an entitlement repository,
24 corresponding to elements 327 and 329, are described in more detail below.

25 The previously described processing of step 327 ensures that valid references for the granting
26 sources and grantee clients are already in place in the entitlement repository 53. An alternate
27 and logically equivalent embodiment is to provide a one step process incorporating a list of
28 initial grantee clients into the metadata update for a new source or a list of granted sources
29 into the metadata update for a new client.

30
31 Step 329 also provides entitlement repository 53 updating for simple entitlements controlling
32 client access to value add services or other resources of the reference data utility. For this
33 sub-case the process is a simple access control list update in the entitlement repository 53

1 using access control techniques well known in the art. An alternate and equivalent
2 embodiment is to combine this step for simple access into the processing of new client
3 metadata to reduce the number of independent processing steps.

4
5 In an advantageous embodiment, data elements 51, source profiles, 52, client profiles, and
6 entitlement repository 53, are stored in the repository 20 reference data utility 1 as described
7 in Fig. 1B. While entitlements have been described as primarily being a grant of entitlement
8 to a particular source for a client or tenant organization, in an alternative embodiment,
9 entitlements can also be associated with value added services indicating that anyone entitled
10 to use the service also derives entitlement to some data or sources associated with the service.
11 Providers of value added service with this property are expected to have obtained
12 redistribution rights to transfer entitlement to data provided to clients on this basis from any
13 sources of the data.

14
15 After appropriate updates have been made to the entitlement repository 53, and to client and
16 source profiles, control flows out of box 210. Processing of the metadata update is complete.

17
18 Fig. 3D illustrates a high-level processing flow for dealing with requests for value added
19 services; an expansion of box 211 in Fig. 2. Within the context of a reference data utility, a
20 value added service is indirectly related to reference data; for example, it uses reference data
21 as input for various data driven computational services or provides a storage service for
22 reference data related business documents. A relationship between a value added service and
23 reference data exists such that it is advantageous to co-locate them in a single logical system,
24 e.g. the utility. Fig. 3D shows two types of value added services: data driven computational
25 services based on reference data and business document storage services.

26
27 Decision element 331 determines whether the received added value request is associated with
28 a data-driven computational service, box 332, or for a business document storage service, box
29 333. If the request is for a data driven computational service, then control flows to outcome
30 box 332. In this case processing flows to decision element 334 which is a test to distinguish
31 between two types of request associated with data driven computational services. The
32 request may contain the specification and executables of an updated or new data driven
33 computational service from a provider which is to be made available to some set of clients of

1 the reference data utility 1. The processing of this, represented by box 335, is to update the
2 registry of available value-add functions with information describing the newly available data
3 driven computational service as indicated by the dashed line from box 335 to data element
4 57. The executables of the function are also stored in the library of data driven computational
5 functions, data element 60, in the repository 20 of reference data utility 1 introduced in Fig.
6 1B as indicated by the dashed line from box 335 to that data element.

7
8 In an advantageous embodiment the input and output datasets of data driven computational
9 service are specified so that they can consume and produce on demand datasets as described
10 below. This means that the provider of a data driven computational service can design and
11 develop it to accept a single format and delivery mode of input data; similarly it will yield a
12 single format and delivery mode of output data. Reference data utility clients can then use on
13 demand dataset processing to connect this with any data to which they are entitled and feed
14 the results of the computation to their own applications without developing custom data
15 formatting and delivery logic.

16
17 The other type of request associated with a data driven computational service is a request
18 from a client for the reference data utility 1 to provide a service instance by invoking a
19 particular data driven computational function with specified input data and returning the
20 produced results as an on demand dataset. This processing is represented by box 336 which
21 shows that both input and output of the data driven computation may be on demand datasets
22 filled either with entitlement managed entity data represented by element 50, or client
23 datasets in the repository 20 of reference data utility 1 represented by element 58. Fig. 4A
24 provides additional detail on the processing of block 336 in a flowchart that shows the steps
25 of a computational added value service flow for a data driven computational service. The
26 preferred embodiment accepts the on demand datasets as an input to a valued added function,
27 an equivalent alternative embodiment allows value added functions to request the creation of
28 an on demand dataset as part of its computation.

29
30 Decision element 337 distinguishes between the processing of three different types of request
31 associated with business document storage services. Boxes 338, 339 and 340 represent the
32 different types of business document storage service requests. Box 338 is a simple request to
33 insert a business document into the business document repository, data element 59, or to

1 update or retrieve a previously stored business document. This processing is further described
2 in Fig. 4B.

3
4 Box 340 represents a request to locate a business document suitable for use with (or to
5 govern) a particular business transaction or to validate the suitability of an identified
6 document for a specific business transaction. An example of this type of business oriented
7 document query is: "does a master swap agreement between counterparties X and Y dealing
8 with financial instruments A and B exist?" This processing to handle such requests is further
9 described in Fig. 4C.

10
11 Box 339 represents a more complex type of business document storage service request,
12 involving choreography of a client's reference data to support the use of one or more stored
13 business document(s) in a particular business operation. This function is described in more
14 detail in Fig. 4D.

15
16 Fig. 3E describes in more detail the processing required to fulfill a general service or report
17 request previously described in box 212 of Fig. 2. Control passes to decision element 350.
18 The request is examined to determine the type of the general service request and routed as a
19 customer service request, box 352, utility report request, box 359, or utility management
20 function, box 353. A customer service request is processed in box 354 after which control
21 proceeds out of box 212. A utility report request gathers data in box 358 after which the
22 requested report is generated in box 360 and then control proceeds out of box 212. A utility
23 management function is executed in box 357, after which control proceeds out of box 212.
24 Dashed arrows connecting box 360 to data elements 54, 55, 56, 62 represent the generation of
25 source, client, function provider and management reports respectively. In an advantageous
26 embodiment these reports are retained in the repository 20 of reference data utility 1 for
27 subsequent access by the owning parties.

28
29 Fig. 4A provides an example flowchart that shows steps in providing a function service
30 instance for a data driven computational service. This flow is an elaboration upon box 336
31 introduced in Fig. 3D, and shows the detailed flow involved in setting up and executing a
32 function service instance for a data-driven computational service. As described with respect
33 to Fig. 3D, requests for data-driven computational services use the same general structure as

on demand dataset requests. Box 636 displays the main aspects of a request specification relevant to computational service requests. These aspects are: 1) the identification of the computational service (function) to be invoked; 2) the specification of input data to be used; 3) the specification of the delivery mode, format, etc. in which the results are to be returned; and 4) the identity of the requester. The identity of the requester is used in several ways; one of which is to check that the requester is entitled to the computational service requested and meets any special requirements imposed by the service. Decision element 638 tests this entitlement using the entitlements repository (data element 53) and the added value function registry (data element 57). If the requester is not entitled to the computation service requested, then processing stops and control exits out of the bottom of box 336.

Upon successful completion of the check, the process formulates an on demand dataset request to collect input data for the requested function instance. This is enabled by the computational service request's use of the same structure as an on-demand dataset request described below. As a result, dataset specification aspects such as selection preference and sourcing preference can be included in the computational service request. The computational service can dynamically formulate a one-time on demand dataset request on behalf of the requester, and submit this request to the data delivery component of the utility 1. As part of this request, the computational service can specify its own preferred format and structure of the data to be returned, removing the restriction to understand a pre-defined data model.

The analysis required to map the original function invocation request to a new sub-request to the data delivery subsystem is shown by box 639. The selection predicate and sourcing preference of the original request are copied to the generated request as is, while the format and delivery mode are specified directly by the computation service to fit preferences for receipt and consumption of input data. The identity of the original requester is also passed on. The generated request is formed and submitted to the data delivery subsystem of the utility, and the response is received as an on demand dataset in box 645. The arrow from box 50 to box 645 represents the movement of an on demand dataset from an entitlement enforcing repository. Because the data is extracted from an entitlement enforcing repository represented by data element 50, the enforcement of entitlements to data based on the identity of the original requester is automatically assured. This provides an additional benefit because it removes the need for computational services to perform their own entitlement management

1 of input data. Input data may also come as an on demand dataset from client datasets as
2 shown by the arrow from data element 58.

3
4 The next step in processing represented by decision element 643, tests to determine whether
5 input data meeting the requirements of the function and the requesting clients entitlements is
6 available. If insufficient data is returned from the previous step, appropriate logging is done
7 and the remainder of the processing is bypassed and control flows immediately out of block
8 336. If sufficient data is available, the functional service instance is executed in box 640.

9
10 Box 641 shows the step of returning the results, in the form of an on demand dataset, to the
11 original requester (client) or saving them in the repository 20 of reference data utility 1 on
12 behalf of the requester as a client dataset (data element 58). In an advantageous embodiment
13 this uses the capabilities of the utility to support on demand delivery of datasets as described
14 in section D below. Because an on demand dataset request specification allows data-marts
15 and client datasets as possible output formats, it is possible to store the results of the
16 computational service in the repository 20. In this case, results are treated as a client-specific
17 data stream, and can be quality assured as described in section C below. The execution of the
18 data driven computational function uses an executable representation stored in the repository
19 20 reference data utility 1 as shown by the arrow from data element 60, the set of data driven
20 computational functions.

21
22 In an advantageous embodiment, the output of the data driven computational function can
23 optionally be stored in an entitlement managed dataset element 50.

24
25 As the last step in the process, any data required for reporting associated with the use of the
26 computational service is generated in box 642. Report types include those delivered to
27 clients (function requesters) and to function providers, represented by data elements 55 and
28 56, respectively. Other report types exist.

29
30 Fig. 4B provides an example flowchart elaborating the steps in handling a request to store or
31 access a business document introduced as box 338 in Fig. 3D. Control flows into this block
32 from the top into decision element 420 which determines whether the business document

1 access request is for inserting a new business document into the store outcome element 421,
2 or for retrieving or updating a previously stored business document, outcome element 422.

3
4 For an insert type, the document to be inserted is received in box 423, along with entitlement
5 information associated with the document. Unlike reference data that arrives from data
6 providers, business documents are received directly from clients of the utility. A document
7 submitted by one client may apply to more than one party, and therefore entitlement for
8 multiple parties may be desirable. During the step shown by box 423, determination of
9 entitlements is made based on the requester, as well as the information contained in the
10 request itself.

11
12 Cataloguing information accompanying the document is received in box 424. This
13 information identifies, describes and classifies the document in the business document
14 repository (data element 59). This information is used for querying, as well as for business
15 document validation processing as described in Fig. 4C.

16
17 An additional set of data choreography rules may optionally be received with the document.
18 Data choreography rules are applicable in scenarios where there is an implied relationship
19 between reference data in the utility and the document being stored. As an example, a
20 document governing allowable mutual fund investments may be linked to financial
21 instruments matching a certain risk profile. Therefore, a rule may be provided for checking
22 whether the risk profile of a financial instrument is within the acceptable bounds described in
23 the business document. Such data correlation rules are optionally received along with the
24 document in box 425. Fig. 4D provides more detail on how data correlation rules are
25 involved in more complex document related processes.

26
27 In step 426, the document and the accompanying cataloguing, validation and data
28 choreography rule information (if any) are stored into the business document repository in
29 data element 59 and entitlement information controlling access to the new document is stored
30 into the entitlement repository, data element 53. An advantageous embodiment uses a
31 method for a repository with entitlement management such as that described below in Section
32 B. Entitlements to documents can be specified at insert time. The process of document
33 insertion may be augmented with manual validation processes to ensure that insert-time

1 specified entitlements comply with security standards of the utility. Alternative embodiments
2 use a standard document management repository solution.

3
4 The functions to update or query documents are shown in the flow starting with outcome
5 element 422. Box 427 represents receipt of document identification or predicate used to
6 select business documents to access. An advantageous embodiment uses a selection
7 preference within an on-demand dataset request, described below in Section D.

8
9 Box 428 is the step of locating the requested document in the document repository and
10 ensuring that the requester is entitled to the document. In an advantageous embodiment,
11 entitlement management is handled with techniques described below in Section B.

12
13 If the operation is an update operation, the updates are applied in box 429. The update is
14 applicable to the document cataloguing information, data correlation rules, and the associated
15 business document. The updated document is stored in the business document repository 59.
16 In this processing step there could also be updates to the entitlements to this business
17 document, giving or removing access for a third party and causing an update in the
18 entitlements repository, data element 53.

19
20 If the operation is a query function, box 430 is the function of returning the requested
21 document and/or associated information for a query function to the requestor. For an update
22 operation an update confirmation message can be returned to the requester. The response is
23 prepared and formatted in a manner consistent with replying to an on-demand dataset request
24 as described below in section D.

25
26 Fig. 4C provides an example flowchart showing the steps in processing a business document
27 validation request. This figure is an elaboration of the processing block 340 first introduced
28 in Fig. 3D which also is shown as a box bounding the control flow in Fig. 4C.

29 Business document validation locates a business document previously saved in the business
30 document store of the utility, which can be used as the reference document for a particular
31 business transaction. In a financial services context, one example is a pair of businesses that
32 agree that transactions of a particular category between them will be executed according to a
33 particular procedure. They document the procedure with a business document which is stored

1 in the utility's document store following the insert or update flow of Fig. 4A. They also
2 document the validation condition, specifying when this procedure is a valid and appropriate
3 procedure, as a set of validation rules appended to the stored business document by step 424
4 of Fig. 4B. In practice for a master agreement governing a trade, these validation rules may
5 be sensitive to the issues such as the amount and value of the traded item, the parties on
6 behalf of which the trade is being executed, and the market and context where the trade was
7 transacted. These validation rules typically refer to reference entities for which the reference
8 data utility is providing values to the transacting parties such as corporate hierarchies,
9 financial instrument definitions and properties, and counter parties etc. It is efficient to store
10 and validate business documents in the reference data utility because of the contained
11 references to other financial entities for which values are needed during validation, and
12 because the document is shared between clients executing a trade. Finally, document
13 validation has to be subject to the entitlements. Validation is done on behalf of a requestor. In
14 order for the request to succeed the requestor has to be entitled to the validation request, and
15 all data and documents required for the validation.

16
17 Processing of a validation request enters through the top of box 340 in Fig. 4C and flows to
18 element 431 where the parameters characterizing the business operation are received from
19 one or both of the requesting parties. These parameters specify characteristics of the business
20 transaction for which an associated stored business document is needed. In the case of the
21 financial trade example introduced above, they include information identifying the items
22 being traded, the amount, the parties executing, the context of the trade and the parties on
23 whose behalf the operation is being executed as indicated above. Using this information, step
24 432 retrieves a set of one or more stored business documents, which are potential candidate
25 matches to be used as a governing document for the specified business operation. The
26 entitlement repository, data element 53, provides the entitlement information and the
27 documents themselves come from the business document repository, data element 59.

28
29 Decision element 438 heads a loop which repeatedly advances to the next candidate
30 document in the list and processes it to determine whether it is a valid match satisfying all the
31 validation rules for this client request. It is possible that the processing of step 432 yielded no
32 candidate documents for validation to which the requesting client is entitled. In that case,
33 control flows via the "No" branch out of decision element 438 and on to box 437. The dashed

1 line from box 437 to box 29 indicated logging of the results. “No matching document” is
2 reported to the client. The same flow using the “No” exit from decision element 438 may
3 also occur after multiple iterations of the loop if all candidates in the initial list have been
4 evaluated and no valid match has been found.

5
6 Step 433 within the loop following the “yes” branch out of decision element 438 advances to
7 the next candidate document. Step 434, also within the loop, evaluates the specified
8 validation rules on that candidate document using context supplied in the request and
9 reference data from the entitlement managed reference data in data element 50. Decision
10 element 435 then tests whether the validation on that candidate document was successful or
11 not. If it was, control flows out of the loop to block 436 which returns the identified current
12 document as the successful match to the requester. The dashed line from box 436 to box 29
13 indicates logging of the results. If the current candidate document did not satisfy the
14 validation rules, control flows back to the head of the loop where decision element 438 tests
15 whether there are more candidate documents available for validation. If this is not the case,
16 no match has been found and this is the reported result of the processing.

17
18 An alternate embodiment always evaluates the validation rules on all candidate documents
19 and returns a list of successfully validated matching documents to the requester instead of
20 returning the first successful match as described above.

21
22 Although the reference data utility stores, locates, and returns a valid business document used
23 to govern the execution of a specific business operation, the actual execution of the specified
24 business operation remains the responsibility of the clients and their trade execution systems.

25
26 Fig. 4D provides a flowchart showing the steps in processing a request to choreograph
27 reference data supplied to a specific business process instance associated with a particular
28 business transaction. This figure is an elaboration of the processing box 339 first introduced
29 in Fig. 3D, also shown as a box bounding the control flow in Figure 4D.

30
31 Reference data choreography supplies current valid reference information supporting a
32 specified business transaction and processing to execute it. The business transaction typically
33 executes on the trade execution systems of the requesting clients, but uses reference values

1 supplied by the reference data utility 1 as reference data choreography. In a financial services
2 context, for example, a trade of common stock may require information about recent dividend
3 payments on the stock and whether they accrue to the buyer or the seller, contact addresses of
4 counter parties to register the transfer with, such as the stock issuer. It may need contact
5 addresses of certificate repositories and other interested parties to complete the transfer, and
6 may need to know the exchange and locality where the stock is traded to understand fee and
7 tax issues associated with the transfer. Much of this information is available to clients of the
8 reference data utility 1 as current values and properties of repository 20 entities. The
9 reference data utility 1 makes entitled information relevant to processing the trade available
10 to one or both parties as part of its reference data choreography processing.

11
12 As shown in step 425 of Fig. 4B, business process data choreography specifications can be
13 attached to each business document stored in the business document repository. The
14 reference data choreography rules specify which values to select from the entitlement
15 managed reference data utility 1 to support a particular business process for which this
16 business document is being used as a guide. Choreography value selection is parameterized
17 with the characteristics of the business transaction being supported. Since a business process
18 typically involves multiple steps with different reference data needed for the different steps,
19 the reference data choreography specification for a given business process takes the form of
20 a set of reference data selections associated with steps in the business process.

21
22 For example, for a business document which is a master agreement governing trade in
23 common stock, parameters for each particular business transaction include the stock symbol,
24 amount traded, trade date and time, trade price, etc. An appropriate reference data
25 choreography step returns the current entitled definition of the stock, its recent dividend
26 history and announcements, counter parties for registering the trade, etc. This information is
27 supplied to the trade execution systems of the utility's clients executing the trade, increasing
28 the reliability, consistency and accuracy of their operations.

29
30 In Fig. 4D, control enters at the top and flows to box 440 where the business process instance
31 parameters, the business document identification and the business process identification are
32 received from the utility client in a request. The business process instance parameters are
33 unique properties characterizing this particular business operation. As described above,

1 examples include the item traded, trade date, trade amount, etc. The client also selects a
2 particular business document to govern the trade execution process. This is done by
3 executing a business process document validation request as elaborated in Fig. 4C or by an
4 explicit selection of a business document by the client or clients. Since there may be multiple
5 business processes associated with a single business document in the store, the specific
6 business process for which reference data choreography is requested is also identified in step
7 440.

8
9 The following step, box 441, retrieves the identified business document from the business
10 document repository and locates the identified business process data choreography request
11 identified by the client. The business document is retrieved from the business document
12 repository, data element 59, after first checking that the requesting client is entitled to access
13 it using information in the request and the entitlement repository, data element 53.
14 Decision element 446 then tests to determine whether a document with matching
15 choreography and to which the requesting client is entitled has been returned in step 441. If
16 not, then no data choreography is possible and control flows out of box 339 reporting this as
17 the outcome of the request. If a business document with matching choreography has been
18 found, control flows on via the yes exit from this test.

19
20 Multiple steps may exist in the data choreography for a specific business process, each
21 parameterized with different input data and each returning a different set of reference values
22 for use in the next step of the process. Element 442 heads a loop. Each iteration of the loop
23 provides the reference data choreography for one step of the identified business process
24 instance. The action of element 442 is to advance to the next process step of the transaction.
25 In element 443 step specific parameters may be received from the requesting client. Element
26 444 uses the step specification provided in the process choreography annotation to the stored
27 business document and following it, retrieves appropriate entitled repository entity values
28 from the entitlement managed repository entity data consistent with the step inputs and the
29 step specification. These values are returned to the requesting client or clients for use in their
30 trade execution system. Appropriate logging and reporting of the delivery is made to a client
31 delivery log as shown by the dashed line from box 444 to data element 29.

32

1 Decision element 445 contains processing to determine whether data choreography for the
2 business process instance is complete or whether there are additional steps to be processed. If
3 the data choreography for the business process is complete, control flows out of box 339. If
4 there are additional steps to be processed, control returns to element 442 and the next step of
5 the data choreography is processed.

6
7 The reference data utility 1 provides reference values to the requesting client or clients. These
8 clients use their own trade execution systems to effect the trade. An advantageous
9 embodiment is to use techniques such as Service Oriented Architecture and Web Services,
10 well known in the art, to enable the efficient interface of different client trade execution
11 systems to the reference data utility 1. Since the reference data values provided in each
12 business process instance step are read-only, minimal state information about the interaction
13 between the client's trade execution system and the reference data utility 1 is needed.

14
15 Dashed lines connecting steps 441 and element 444 with the entitlement repository 53, the
16 entitlement managed repository entity data 50 and the business document repository 59, show
17 where these sources of data are used.

18
19 The services for validating and providing reference data choreography are useful, but
20 optional, extensions of the basic capability to store and access business documents in the
21 reference data utility store.

22
23 An alternate embodiment of business document function is to provide clients with alerts
24 when there is a change in reference data which affects the meaning or usefulness of their
25 documents in the business document repository. For example a change in corporate
26 ownership hierarchy may affect a set of business documents - specifically master agreements
27 governing transactions may need to be reviewed when there are changes in the hierarchy of
28 corporate entities which could be participants. Using the on demand dataset capability, the
29 reference data utility 1 can monitor changes affecting specific sets of business documents on
30 behalf of clients and deliver affected document identifiers to them when such changes occur.

31
32 Fig. 5A describes the types of reports that the utility 1 can generate for clients, data sources,
33 providers of value-add functions, regulators and internal management. A simple hierarchy

1 starts at box 502 with report types. The utility 1 can provide multiple types of reports; reports
2 to clients, box 505, reports to data sources, box 511, reports to function providers, box 519,
3 reports for regulators, box 520, and internal reports used to manage the utility, box 518.

4
5 Reports for regulators 520 are defined by the relevant regulatory agencies. Internal reports
6 518 are defined as needed by the utility operator.

7
8 Client reports include, but are not limited to, delivery log reports, box 506, source utilization
9 reports, box 507, source accuracy reports, box 508, reports on source timing, box 509, service
10 level reports, box 510, and reports generated for customers which they have to give to
11 regulators, box 504. Clients may be regulated by different agencies than the utility and as
12 such their reporting requirements may be different. These reports are defined by the
13 regulatory agencies and generated as needed.

14
15 The utility generates three categories of reports for data sources; accuracy reports, box 512,
16 timing reports, box 513, and quality and usage reports, box 514. These reports are designed to
17 help the source vendor improve and manage their data quality by assisting in identifying the
18 issues that are critical to the source vendor's customers.

19
20 Function provider reports in box 519 provide information gathered by the reference data
21 utility 1 on usage of the provided functions to support assistance from the reference data
22 utility 1 in client usage accounting and billing.

23
24 Fig. 5B gives an overview of the utility management functions represented by box 503.

25 Utility management functions are divided into three broad categories; performance, ellipse
26 515, service level agreement, ellipse 516, and infrastructure, ellipse 517. The performance
27 function allows the utility operator to monitor performance based on metrics defined by the
28 operator. Monitoring enables the utility to manage performance manually, automatically or
29 through a combination of both. Service Level Agreement (SLA) functions allow the utility to
30 monitor its performance against its SLA commitments and manually or automatically manage
31 its operations to improve utility performance as evaluated by the SLAs. The infrastructure
32 function supports the efficient management of the processor's storage, software and other
33 information technology used by the reference data utility 1 or its operations.

Fig. 6 addresses the geographical dispersion and high availability issues affecting a multi-source multi-tenant reference data utility.

Boxes 601, 602 and 603 each represent a utility site located in different cities around the world; in this example New York, London and Singapore, respectively. The technique can be applied to any number of sites in any set of locations. Each of these sites has processing capabilities of a utility, corresponding approximately to the capabilities represented by reference data utility 1 in Fig. 1A. A data acquisition and quality enhancement component, box 19 as first introduced in Fig. 1A and a client data delivery component, box 21, are shown at each site. The high quality of data values in each repository 608, 609, 610 is maintained by a pool of human experts with deep business knowledge of relevant topics; these experts make judgments about arriving values to ensure that data delivered to customers is of the highest quality. Therefore, the effectiveness of the utility depends on availability of the best experts on each topic to process information on that topic in a timely way at the lowest cost. It is assumed that experts on regional issues will be located in proximity to the region. Ellipses 605, 606 and 607 represent the human pools of experts providing these quality assurance services on arriving data, and associated customer services. The function of each of these pools corresponds to ellipse 37 in Figure 1A. Similarly elements 608, 609 and 610 are site specific versions of the repository 20 of reference data utility 1 in Fig. 1A. Fig. 6 expands the utility concept as described in Fig. 1A, by including multiple sites. In a multi-site utility, data quality enhancement for a particular subtopic need be performed at only one site; this task can be assigned to the site where it is performed most efficiently. Hence, topics or subtopics are partitioned and each is assigned for primary quality assurance to a site, as represented by boxes 601, 602 or 603.

Links 604 represents a high speed, world-wide communications fabric connecting the geographically dispersed sites. This capability ensures that the multi-site utility is able to operate as a single logical service, making data available to clients regardless of where they or their subscribed vendor sources are connected, and ensuring that backup service is available for utility capabilities from another site should a site be disabled. Although reference data for a topic is cleansed at a selected primary site, in an advantageous embodiment, the cleansed entity data on each topic is then copied to all sites for ease and speed of delivery to clients. Also, updated entitlement repositories are maintained at each

1 site, at least covering entitlements of clients attaching at that site. Hence all sites are involved
2 in cleansing; each item of arriving data is acquired and quality enhanced once and all entity
3 data is available to all entitled clients via local repository access with local entitlement
4 enforcement. Use of a guaranteed messaging system for propagating cleansed data from the
5 primary site to other sites, assures that updates are propagated to remote sites without risk of
6 data loss. In an alternate embodiment, cleansed data and entitlements are stored at a more
7 restricted number of sites; requests to retrieve and deliver reference data must be sent to one
8 of the sites where the data is located. One form of this restriction is to retain and store
9 cleansed data only in its primary cleansing site. There are availability, resiliency and
10 redundancy advantages in storing each item of data at a plurality of sites, prompting
11 intermediate alternate embodiments where each data item is stored at more than one, but not
12 all sites.

13
14 In the example of Fig. 6, data sources S1, S2, S3, S4, S5 and S6, represented by circles 620,
15 621, 622, 623, 624 and 625, respectively, each connect to one of the utility sites. There is an
16 assumption that high speed, world-wide communications (connecting links 604) allows data
17 from each source to be distributed wherever needed for input processing, quality assurance or
18 storage in a repository. Similarly, clients C1, C2 and C3, (represented by circles 611, 612 and
19 613) are attached at repository site A, clients C4, C5 and C6 (represented by circles 614, 615
20 and 616) are attached at repository site B, and clients C7, C8, C9 (represented by circles 617,
21 618 and 619) are attached at repository site C. This set of example client and source
22 attachments illustrates properties of the multi-source multi-tenant reference data utility.

23
24 The reference data utility treats each connecting client as an independent logical entity with
25 specific entitlements to which data can be delivered. A single corporate tenant may have
26 associated with it clients which connect at a plurality of reference data utility sites. The
27 higher level corporate ownership may be reflected in entitlement structures, and in client
28 profiles, but does not alter the methods for delivering retrieved data to each connecting client
29 described in this method. For the purposes of delivering on demand data sets and executing
30 value add functions, the utility treats each local client as an independent owner of a client
31 profile and submitter of requests to the utility for retrieval and delivery of data. For the
32 purposes of accounting, entitlement tracking, service level reporting, contract management
33 and authorization management, the utility can maintain awareness of hierarchical

relationships associating connecting clients with possibly geographically dispersed corporate entities to which they belong.

Each client C1, C2, ... C9 attaches at a single site but has access to all reference data in the dispersed reference data utility to which they are entitled regardless of the site used to provide quality assurance on those values, the site of the connection points for data sources to which that customer is entitled, the site of primary storage for that data (when data partitioning is used), or the failover or backup site providing master storage and update of values for that topic or subtopic during a temporary failure of a master site.

Repositories 608, 609 and 610 represent reference data utility repositories (corresponding to the logical capabilities of repository 20 in Fig. 1A) maintained at each utility site. The repository at each site is aware that it is the master (source of) for some reference topics. The results of data gathering and quality assurance on those topics are subsequently propagated to remote sites from that site. For other reference topics, this site will receive and hold values from whichever of the other repository sites is acting as the master. In an alternative embodiment, the data is replicated and enhanced at all sites. In another alternative embodiment the data can be partitioned between sites and each data element stored at a single site only. Replicating the data to all sites provides better availability and ensures that each site is responsive to locally attached customers requesting data. It may be sufficient for arriving raw data logs and customer delivery logs to be stored only at the repository site where data is received and quality assured or where a logical customer is locally attached. In an alternative embodiment, where data is partitioned and held at a small number of sites, the differences in the assignment of storage and data quality assurance responsibilities makes each repository site distinct and enables each repository, though functionally similar, to hold different data.

This concludes the description of the flow diagrams for section A describing the overall reference data utility and associated value add functions. In preferred embodiments workflows are used to implement the process and flows described herein. Alternative embodiments use script, discrete distributed process, or a mixture of all of these. Any suitable mechanism or programming language is used to implement the flows and processes described herein.

1 B. General Structure and Method of Operation of the Repository

2 This aspect of the invention is directed to a multi-source multi-tenant data repository (herein
3 referred to as “repository”) with entitlement management based on source tracking of
4 reference data values and to a method for operating it. Such a multi-source multi-tenant data
5 repository with entitlement management is an important component of a multi-source multi-
6 tenant reference data management service or of utility 1, described above. It is also useful in
7 other contexts. The multi-source multi-tenant data repository manages and provides
8 permanent storage for repository information elements, associated metadata, entitlements,
9 value add functions and documents, and may function as repository 20 described above.

10
11 Throughout we illustrate aspects of the invention with examples of financial reference data
12 such as descriptions of financial instruments, counterparties, corporate legal entity hierarchies
13 and corporate action events. Reference data in these categories is widely used in financial
14 markets. The methods of the invention are also applicable to provide and support other
15 classes of reference data with similar characteristics. In particular a multi-source, multi-
16 tenant entitlement repository with source based entitlement management is useful wherever
17 there are many sources and many tenants with independent source based entitlements needing
18 to search and retrieve values to which they are entitled but, in general, not needing to update
19 the data directly.

20
21 The repository also includes data retrieval, access and query mechanisms available to
22 requesters (for example tenants, or agents acting on their behalf). Advantageous innovations
23 of the repository component that distinguish it from a standard database are:

24 - the repository incorporates the ability to store multiple versions of attributes (versioned
25 attributes), where each version is deemed distinct based on value, metadata, temporal
26 information or sourcing information;

27 - the repository retains full information about the history and sourcing of all information
28 elements. The history includes the following aspects:

29 - all events pertaining to the information element in question;

30 - all sources and agents of such events; and

31 - chronological order of such events.

32 - the repository maintains source based entitlement information on all authorized requesters
33 and on all entitlement grants from particular sources to particular requesters; and

1 - the repository incorporates the ability to service requests for the information it includes
2 based on selection and sourcing preferences of the requester, and source access driven
3 entitlements.

4
5 The data in the repository is organized to allow shared access paths. Access paths and
6 indexing are available to all requesters to select reference item values of interest and they
7 provide client-specific entitlement-based access to reference data values.

8
9 The repository allows individual requesters to specify their preferred source for retrieved data
10 at the field level. This preference will be used in choosing between available values from
11 different sources entitled to the requester.

12
13 All of the above capabilities are provided in an environment in which the security and
14 privacy of customer and vendor actions are maintained. No customer or data vendor is able to
15 discover information about another's data, queries or other actions by the repository to
16 support them.

17
18 The method is described herein as it applies to reference data used by Financial Services
19 businesses. This method for forming and organizing a multi-source multi-tenant data
20 repository of reference information with entitlement management based on source tracking of
21 reference data values has many other possible areas of application. Access to consumer
22 credit information, government regulation and registration information, and
23 telecommunications usage information are three additional examples where the method has
24 use. Characteristics of contexts where the method has use and of reference data are: (1) the
25 information comes from many sources; (2) there are multiple users, potentially in
26 independent organizations, that need access to the same information but potentially with
27 different source entitlement rights; (3) the referenced information is accessed by users largely
28 in read-only mode except when they participate in correcting invalid values; (4) high quality
29 timely information is both valuable and complex to gather, hence the efficiencies from a
30 utility approach, shared infrastructure and shared data quality enhancement provide
31 significant benefit; and (5) entitlement enforcement and privacy management must be
32 provided by such a utility. Although the invention is described in the context of financial
33 services reference data, which is one important area of application, the approach revealed

1 herein, enabling an effective utility to provide data access meeting the requirements above,
2 has value in any context with these requirements.

3
4 When the repository is being used in the context of a reference data utility it corresponds to
5 element 50, the entitlement managed entity data, appearing as part of the reference data
6 utility repository 20 in Fig 1B.

7
8 Fig. 7A shows an example of a method for managing information and associated source
9 based entitlements in a multi-source multi-tenant data repository. This figure represents a
10 high level overview of the advantageous processes needed to form, maintain and operate the
11 repository. In Fig. 7A, box 1100 represents the overall method. Within it, box 1101
12 represents the initial step of forming the repository with the necessary information element
13 structures in place (described in detail in Fig.8A, 8B, 8C, 8D). In addition to these, the
14 repository is used to store other items that reside in a data store. These additional items are
15 business (value added functions, business documents, etc.) or functional/operational (rule
16 sets, log records, etc.) in nature as was described in the description of box 20 in Fig 1B.

17
18 Box 1102 is the function of inserting arriving information elements into the store, annotating
19 each element with annotations describing its evolutionary history. . These annotations are
20 known as evolutionarily tracked source data tags (ETSDTs), and can be associated with any
21 information element (or set of elements) in the repository. Each event (the term “annotation”
22 is also used synonymously throughout this document) in an ETSDT effectively corresponds
23 to some action performed upon the information element being described and corresponds to a
24 distinct version of that information element. Each event within an ETSDT carries important
25 information, in particular, the source, or sources, of the event (a source can be a single-source
26 or a multi-source process, as well as an atomic source such as “original document”), the agent
27 who performed the event, event identifier information, timestamp information and descriptive
28 information about the event. Other attributes are possible. Recording full sourcing
29 information in this way provides full traceability to all sources that contributed to the creation
30 of the information element value. This full traceable history is a advantageous enabler of a
31 multi-source multi-tenant data repository wherein the intellectual property rights of source
32 providers and privacy rights of data consumers can be protected. See Figures 8A, 8B, 8C and

1 8D for examples of information elements and associated ETSDTs. Arrow 1110 represents
2 information elements arriving as input to the insertion step of box 1102.

3
4 Box 1103 represents the repository's ability to maintain source based entitlement information
5 about authorized requesters of repository information and data sources to which they are
6 entitled. For example, in a financial reference data repository, a record specifies that
7 repository tenant A is entitled to financial instrument data from source providers A and C
8 only (whereas the repository may include data from providers A,B,C, D, E, F, and G). Arrow
9 1111 represents updates in entitlement information received as input and handled by the
10 entitlement maintaining process of box 1103. One possible choice for an embodiment of box
11 1103 is for updated entitlement information to be stored in the multi-source multi-tenant
12 repository; an alternate embodiment is to maintain entitlement information following the
13 processes described herein but storing the updated entitlement information in a separate
14 repository.

15
16 Box 1104 represents the ability of the repository to use ETSDTs together with source based
17 entitlements in a process that provides controlled access to the information included in the
18 repository. This process takes into consideration various sourcing and selection preferences
19 of the requester. For instance, in a financial reference data repository, this process is able to
20 respond to a request to return information on all stocks in an interest list A from all available
21 sources. In this example the process would identify the requester, retrieve their entitlements,
22 and then select and return the information set forming the intersection of the request
23 specification and the entitlement restrictions.

24
25 Arrow 1112 shows retrieval requests arriving as input to the processing of box 1104; arrow
26 1113 shows retrieval responses being returned as output for this processing.

27
28 Thus, the present invention includes a method for sustaining a multi-source multi-tenant data
29 repository. The step of sustaining including the steps of: forming the multi-source multi-
30 tenant data repository to include information elements from a plurality of sources, describing
31 at least one referred entity; annotating a plurality of elements from the information elements
32 in the multi-source multi-tenant data repository with sourcing information; maintaining
33 information about entitlement of requesters to information elements based on the sourcing

1 information; and responding to at least one request from at least one requester to return a set
2 of information elements based on requester-specified selection predicates and sourcing
3 preferences and subject to the entitlement of the at least one requester.

4
5 In a financial market example used herein, the method is for sustaining a financial multi-
6 source multi-tenant data repository. The step of sustaining includes the step of forming the
7 financial multi-source multi-tenant data repository to include information elements from a
8 plurality of sources, describing at least one referred entity. Consider sources feeds from
9 Vendor A, Vendor B, and Vendor C . The method also includes the step of annotating a
10 plurality of elements from the information elements in the multi-source multi-tenant data
11 repository with sourcing information. Examples of sourcing information include that a
12 specific set of values defining the common stock of company A were received from the
13 Vendor B feed in a data record with record identifier R received at time T. It also includes the
14 step of maintaining information about entitlement of requesters to information elements
15 based on the sourcing information. Examples of this include that client C is entitled to
16 receive data from Vendor A and Vendor C feeds but not from the Vendor B feeds. It also
17 includes the step of responding to at least one request from at least one requester to return a
18 set of information elements based on requester-specified selection predicates and sourcing
19 preferences and subject to the entitlement of the at least one requester. Examples of this
20 include returning to client C the current entitled recommended definition of the common
21 stock of company A.

22
23 Fig. 7B is an alternate more detailed control flow of an advantageous embodiment for the
24 method showing how each individual arriving input, i.e. information element, update to
25 entitlements or retrieval request, is handled when it arrives at the previously formed
26 repository. This representation shows that the insertion of new annotated information
27 elements, updating of entitlement information and responding to retrieval requests can be
28 interleaved.

29
30 In Fig. 7B, box 1100 again represents the overall method. Control enters from the top. The
31 initial step is to form the repository establishing the essential data structures with box 1101 as
32 described above. At this point the repository is ready to receive inputs. The inputs are
33 represented by the arrows 1110, 1111, 1112, representing arrival of new information

1 elements, entitlement information updates and requests for information retrieval, respectively.
2 Box 1105 is the step in the control flow where all of these arriving inputs are first handled. It
3 heads a loop from box 1105 to box 1114; each iteration of this loop will handle one arriving
4 input.

5

6 The first control flow step in processing an input is to determine its type. This is done in the
7 decision element 1106. The method handles three primary types of arriving action prompt: a
8 new or updated information element, an entitlement update and a request for information.

9 These outcomes from decision element 106 are handled by the paths headed by boxes 1107,
10 1108, and 1109 respectively. The processing of a single arriving information element is
11 handled by a control instance of the insertion and annotation process in box 1102. This
12 processing was discussed when box 1102 was first introduced above in Fig. 7A. The
13 processing of a single arriving update to entitlements is handled by a control instance of the
14 “maintaining source based entitlements” process represented by box 1103. This processing
15 was discussed when box 1103 was first introduced above in Fig. 7A. The processing and
16 response to single request for repository information is handled by the “responding to
17 requests to return information elements” process represented by box 1104. This processing
18 was discussed when box 1104 was first introduced in Fig. 7A.

19

20 After completing the processing of an arriving information element, entitlement update or
21 request for information, a choice is made in decision element 1114 whether to return to the
22 head of the loop to handle more inputs. Under usual conditions when the repository is not
23 shutting down the Yes branch will be taken and control flows back to the top of the action
24 loop awaiting the next arriving action prompt. Repeated instances of this action loop result in
25 additional information elements being added into the repository with annotations, additional
26 entitlement updates being received and saved, and additional requests for retrieval of
27 information stored in the repository being served.

28

29 The above flow is a logical control flow describing the method. Using well understood
30 transaction, database and computer concurrency techniques, an advantageous embodiment of
31 the method is able to handle multiple actions from different sources and requesters
32 concurrently.

33

Fig. 8A shows an example of a conceptual organization of the repository's top level information elements. Box 1201 represents the overall repository, also represented generally as 20 in the discussion above. At the top level the repository includes a list of repository entities as represented in box 1202. Example repository entities ENT1, ENT2, and ENT3 within this list are represented by boxes 1203, 1204, and 1205, respectively. A repository entity (e.g. box 1203) is a collection of information all of which describes a single referred entity. For example, in a financial reference data repository, a repository entity might correspond to "common stock of company X".

Each entity has associated with it an evolutionarily tracked source data tag (ETS DT). In the advantageous embodiment, ETS DTs are also attached as annotations to other lower level information elements in the repository. An ETS DT stores event information associated with the information element which it annotates and essentially chronicles the evolutionary history of the information element. This includes information describing: creation of the element, modification of its properties, creation of versions, etc. Each event stored with an ETS DT carries various information (identifiers, event descriptions, user IDs, timestamps etc.), but most importantly each event has a source (or sometimes multiple sources) and, if appropriate, an agent. The resulting availability of a fully sourced history for each information element is an enabler of the multi-source multi-tenant aspects of the repository. Information elements 1206, 1207, and 1208 represent the ETS DTs attached as annotations to example entities ENT1, ENT2, ENT3 respectively. At the entity level, the ETS DT records the information and associated quality enhancement actions, which prompted the creation of this repository entity.

Fig. 8B shows an example organization for the information of an entity in the repository showing the contents of the entity in more detail. Box 1203 is redrawn since it was already introduced as entity ENT1 in Figure 8A. The previously introduced entity ETS DT for ENT1 is also redrawn in Fig. 8B attached as an annotation to ENT1 represented as data element 1206.

Each repository entity includes a list of entity properties represented as box 1209 and a list of entity item instances represented as box 1216. Entity properties are additional information about the entity that can include metadata information and business information about the referred entity that is not necessarily associated with a paid, or otherwise restricted source.

1 Hence, properties could be internal identifiers, non-vendor owned classification information,
2 etc. Normally, information stored within properties is made available to requesters in an
3 unrestricted fashion and, as such, is used to construct indexes and to locate and select entities
4 through shared access paths available to all tenants of the repository. Examples of properties
5 of a repository entity, which refers to a financial instrument include: the full name of the
6 instrument, identification as a stock or a bond, the industrial sector of the issuing corporation,
7 etc. These properties are either public information or otherwise equally accessible to all
8 tenants due to some business arrangement with tenants and/or data providers. If a property
9 requires restricted access for whatever reason it should be represented as a versioned attribute
10 instead.

11
12 Example repository entity ENT1 is shown with three entity properties P1, P2, and P3
13 represented by boxes 1210, 1211, and 1212 respectively. In this example, each entity
14 property has annotations within the parent entity ETSDT (box 1206) relating to them. An
15 advantageous embodiment places property annotations within the parent entity ETSDT. An
16 alternative implementation could have separate ETSDTs associated with the properties.

17
18 A repository entity includes a list of item instances. Each item instance gathers together and
19 includes a set of all attribute values for the parent entity provided by a single, common
20 sourcing. One common sourcing could be that all data in the item instance originated from a
21 single source dataset provided by one source (e.g. Data Vendor A). Another common
22 sourcing is that the data in the item instance was provided by a single identified item instance
23 process (e.g. Value Comparison Process B). Distinct support for both types of sourcing is
24 important because in the case of multi-source data enhancement processes, both the item
25 instance process and the data sources contributing to that item instance process play a role in
26 determining entitlement. This is further described in the entitlement enforcement processing
27 description of Fig. 11E.

28
29 To further elaborate on item instance processes, an item instance process is any process that
30 is used to create, update or review item instances. The concept of an item instance process
31 covers many common methods of creating and working with item instances. Examples of
32 item instance processes include: getting a feed/dataset of items from a source and applying
33 validation, normalization and cleansing to the dataset; employing cross-source processes to

1 compare information from several sources and selection of a preferred value based on this
2 comparison; employing cross-source processes to create composite values that include
3 attributes from multiple sources; and running an algorithmic value enhancement process
4 against values provided by another source. Each such distinct process generates a separate
5 item instance that is stored under the appropriate repository entity. It's possible to have
6 composite item instance processes – as such, both “Normalized” and “Normalized, and
7 Single Source Cleansed” are valid item instance processes where the former is a simple item
8 instance process and the latter is a composite one, comprising of a normalization process and
9 a single source cleansing process. Whether only a single source or multiple sources of
10 information are employed during processing is an advantageous characteristic of an item
11 instance process.

12
13 Box 1216 represents the list of item instances included in example repository entity ENT1 in
14 Figure 2A. Boxes 1217, 1218, and 1219 represent example item instances in this list, ITM1,
15 ITM2, and ITM3 respectively. Each of these has an associated ETSTD attached to it as an
16 annotation represented in the figure as rectangles 1220, 1221, and 1222 respectively.

17
18 In the context of a financial instrument reference data repository, possible examples of item
19 instances for the entity representing “common stock of company X” include: (1) data on this
20 instrument provided by Vendor A, (2) data on this instrument provided by Vendor B or (3)
21 data on this instrument obtained from a repository service which compares data from multiple
22 sources and selects a recommended value from these possibilities.

23
24 Note that an alternative embodiment may have a different scope for the various ETSDTs
25 described (for instance, it is possible to have an implementation with a single logical ETSDT
26 for entities and item instances, reflecting events in the history of both information elements).
27 However, any such alternative implementation logically corresponds to the structures
28 described herein.

29
30 Fig. 8C is an example organization for the information of an Item Instance showing its
31 content in more detail. Box 1217 represents an expanded view of the example item instance
32 ITM1 originally introduced in Fig. 8B. Data element 1220 represents the item instance's
33 ETSTD previously described in Fig. 8B. In Figure 8C, item instance ITM1 includes a list of

1 versioned attributes represented as box 1223 and a list of properties represented as box 1230.
2 The properties have annotations related to them stored in the ETSDT of their parent item
3 instance (box 1220).

4
5 Each versioned attribute in the versioned attribute list includes a set of attribute values
6 characterizing the parent repository entity with values provided by the source or item instance
7 process associated with the parent item instances. For the previously introduced example of a
8 repository entity with information about “common stock of company X”, examples of
9 versioned attributes include (1) current price, (2) exchange where traded, (3) announced
10 dividend accrual date, and (4) announced dividend amount.

11
12 In Fig. 8C, for item instance ITM1, versioned attributes VA1, VA2, and VA3 in the
13 versioned attribute list are represented by data elements 1224, 1225 and 1226 respectively.
14 Each of these versioned attributes has an associated ETSTD attached to it as an annotation,
15 represented herein as data elements 1227, 1228, 1229.

16
17 Item instances also have associated properties that are available for use by requesters to
18 access information stored in the repository. Item instance properties P4, P5, and P6 in ITM1’s
19 property list are represented by boxes 1231, 1232, and 1233, respectively. An important
20 example of an item instance property is the unique item instance process identifier or source
21 dataset identifier characterizing the source of information in the item instance. Item instance
22 properties are also information elements and have annotations within the item instances
23 ETSDT’s relating to them.

24
25 Fig. 8D shows an example organization for the information of a versioned attribute showing
26 its contents in more detail.

27
28 The enlarged box 1224 with its attached versioned attribute ETSTD, represented as data
29 element 1227, includes this expanded view. It shows that a versioned attribute consists of a
30 list of attribute values. Box 1237 represents the list of values for example versioned attribute
31 VA1 as attribute values V1, V2, V3 in boxes 1238, 1239, and 1240, respectively.

1 Attribute values are the lowest level of information element and represent the atomic pieces
2 of business data from which higher level versioned attributes, item instances and repository
3 entities are composed. Multiple values of attributes exist within an item instance for one of
4 the following reasons: (1) several collection and quality enhancement actions have been
5 applied to the original source data leading to several viable values, (2) multiple values have
6 been supplied by a single source for this attribute, or (3) the given item instance represents
7 data produced by multi-source item instance process, and alternate values for the attribute are
8 available from different sources.

9
10 When item instance processes modify an attribute more than once, each modification creates
11 a new value (version) of the versioned attribute. The structure that allows detailed tracking
12 of these changes is the versioned attribute ETSDT, which includes annotations pertinent to
13 each attribute value. Each annotation is directly associated with a specific attribute value.
14 The information stored in the ETSDT allows historical traceability of every attribute
15 modification and, most importantly, includes information about the source(s) and agent(s) of
16 such modifications. This knowledge is later used to decide whether the value can be
17 provided to a specific requester.

18
19 To elaborate on the financial instrument example (using common stock of company X), item
20 instance process P is an automatic cross-source comparison and value selection process
21 which creates composite item instances. An employee employed on behalf of a reference
22 data repository is responsible for reviewing and correcting (as necessary) the resulting
23 composite item instances. The first time that process P is executed, a new item instance, I,
24 would be created under the repository entity representing common stock of company X. A
25 property on that item instance indicates that process P is the item instance process producing
26 this item instance. Since an item instance is composed of attributes, for a given attribute A
27 within I, process P includes, for example, the comparison and review of five attribute values
28 V1, V2, V3, V4 and V5 provided by different sources (data providers). At the completion of
29 process P, value V3 of attribute A is selected. In this example, value V3 would exist as a
30 separate value (version) within the versioned attribute A, and would have a corresponding
31 annotation in the versioned attribute level ETSDT, stating that V3 matches the value provided
32 by data provider DP1 (source 1) and data provider DP5 (source 2), and was further confirmed
33 based on review by data cleanser DC1 (agent) who, in turn, based the decision on review of a

1 public document of Company X (source 3). As evidenced, this sourcing information can be
2 complex, given the complicated potential item instance processes. An innovation of the
3 repository is the ability to carefully keep track of all such sourcing history and then use it as a
4 basis for responding to request for data within the confines of requester entitlements
5 (described in Figs. 11A, 11B, 11C, 11D and 11E.
6

7 In addition to storing repository entities with associated properties, item instances, versioned
8 attributes and attribute values, the repository is used to store other objects such as value
9 added functions and business documents. Entitlement tracking for these objects is needed as
10 well, and it is possible to handle them entirely using the data structures described above.
11 However, if the level of versioning and multi-sourcing for these objects is significantly
12 simpler than the method was designed to provide, an alternate, and advantageous,
13 embodiment is to store each such object in a separate list in the repository, with associated
14 ETSDTs recording source and creation history, but storing all the object information in a
15 simple entitlement managed value box. Such stored objects still have generally accessible
16 properties at the top level enabling requesters to access them readily.
17

18 As in Fig. 8A, it should be noted that an alternative embodiment may elect to have a different
19 scope for the various ETSDTs described (e.g. have separate ETSDTs for item instance
20 properties). However, any such alternative implementation logically corresponds to the
21 structures described herein.
22

23 Fig. 9 expands box 1102 from Fig. 7A labeled "inserting information elements with sourcing
24 annotations," providing more detail about the sample control flow for an advantageous
25 embodiment of this box. Multiple control flows exist based on the kinds of events and kinds
26 of information elements being updated, however, they all follow the same general principle.
27 For purposes of illustration, four processes are chosen: creation or updating of a new entity,
28 creation or updating of a new entity property, creation or updating of a new item instance and
29 creation or updating of a new attribute value.
30

31 Control flows into box 1102 in Fig. 9 when a new information element event arrives at the
32 repository. The new information element to be inserted into the repository is available as an
33 input parameter to the flow of Fig. 9. Box 1301 represents acceptance of the input event.

1 Decision element 1302 is a test to determine the type of the new information element
2 presented for annotation and insertion into the repository. Detailed flows are provided
3 corresponding to creation or update of a new entity, creation or update of an entity property,
4 creation or update of an item instance, and a new or updated value for an existing versioned
5 attribute. These flows are represented by the outcome paths from decision element 1302
6 leading to boxes 1303, 1306, 1310 and 1314 respectively.

7
8 The Fig. 9 control path starting with box 1303 shows an example of a detailed flow for the
9 creation of a new repository entity or update of a property of an existing repository entity. In
10 the context of the financial instrument example this occurs when the repository starts keeping
11 information on a new financial instrument or changes a property such as the "industry
12 grouping" in which this instrument is classified.

13
14 Box 1303 represents the identification that the arriving information element defines a new
15 entity. Box 1304 is the action of adding the new entity into the repositories entity list. Box
16 1305 is the action of creating the annotating entity ETSDT for the newly inserted entity. The
17 dashed line joining box 1305 with data element 1206 shows that the updates are applied in an
18 entity ETSDT as introduced in Fig. 8A.

19
20 The Fig. 9 control path starting with box 1306, shows an example of a detailed flow for
21 updating or creating a new repository entity property. In the context of the financial
22 instrument example discussed above, this occurs when some classification of the instrument
23 is first known or changed, such that it is associated with the transportation industry.

24
25 Box 1306 labels that we are on the new entity property path. Box 1307 is the step of locating
26 the parent entity described by this property. Box 1308 is the step of inserting the received
27 property value into the property list for that entity or updating a previous value. Box 1309 is
28 the step of annotating this new property with an ETSDT recording its source and other events
29 in the path of creating a quality assured version of the received information. The dashed line
30 to data element 1213 shows that this annotation is stored in the repository as an entity
31 property ETSDT as described in Fig. 8B.

1 The Fig. 9 control path starting with box 1310 shows an example of a detailed flow for
2 creating a new item instance for an existing repository entity. In the context of the financial
3 instrument example discussed previously, creation of a new item instance for a repository
4 entity whose referred entity is a corporate bond or common stock occurs when either a data
5 provider, a source of information or an item instance process, such as a multi-source data
6 quality enhancement process associated with the repository itself, starts providing attribute
7 values for this bond or stock.

8
9 Box 1310 represents the identification of a new item instance for an existing repository
10 entity. Box 1311 represents the identification of the location of the appropriate parent
11 repository entity to which the new item instance pertains. This is done on the basis of the
12 referred entity or, if no repository entities currently exist for the referred entity, a process for
13 creating a new repository entity is triggered. The flow continues after the proper parent
14 repository entity has been located or created. Box 1216 in Fig. 8A shows that the list of item
15 instances is a top level data structure in each repository entity. Box 1312 represents creation
16 of a new item instance in this list using the provided item instance information or, if the
17 arriving element is a property update to an existing item instance, applying this change. Box
18 1313 is the action of either creating a new item instance ETSDT or annotating the property
19 change in an existing one. A new ETSDT records the creation of the item instance, and
20 serves as the first annotation in the history of this item instance. The dashed line connecting
21 box 1313 with data element 1219 shows the association between this update action and item
22 instance ETSDT introduced in Fig. 8A.

23
24 The Fig. 9 control path starting with box 1314 shows an example of a detailed flow for
25 creating or updating an attribute value in an existing item instance of an existing repository
26 entity. In the financial instrument example discussed earlier, examples of processing new
27 attribute values include when a particular source or item instance process provides new
28 values for an attribute of the instrument, e.g., exchange where traded, maturity date or rating
29 of a bond, or the date of accrual and amount of a dividend payment on a common stock.

30
31 Box 1314 represents identification of the new attribute value for an existing item instance of
32 an existing repository entity. Box 1315 represents the identification of the location of the
33 parent repository entity to which the new attribute value pertains. This is done on the basis of

1 the referred entity. Box 1316 represents the identification of the location of the parent item
2 instance to which the new attribute value pertains. This is done on the basis of the item
3 instance process which triggered the input event. Box 1317 represents the identification of
4 the location of the specific versioned attribute to which the new attribute value pertains. Box
5 1223 in Fig. 8B shows a list of versioned attributes to be a top level data structure of an item
6 instance. In the financial instrument example discussed previously, information such as the
7 exchange where traded, coupon payment details, rating, dividend amount and data are distinct
8 versioned attributes of the subject financial instrument. Box 1318 represents addition of the
9 new or updated value to the versioned attribute. Box 1237 in Fig. 8D shows that a list of
10 included values is a top level data structure of a versioned attribute in the context of
11 versioned attribute VA1.

12
13 Box 1319 represents the annotation of the new value within the ETSDT of the versioned
14 attribute. The sourcing information included in the annotation exactly identifies the source(s)
15 of the new value. The sourcing information is also a convenient place to store other
16 information related to this event, such as: (1) specific documentation of the reasons for
17 having the new value (e.g. the value was flagged for review by the cleansing engine), (2)
18 specific documentation of research or validation actions taken (e.g. looked up the value in
19 source A), (3) agent of the change (for instance, an employee tasked with reviewing values),
20 etc. The dashed line connecting box 1319 to data element 1231 shows that the data object
21 impacted by this tagging process is a versioned attribute ETSDT as introduced in Fig. 8D in
22 the context of the ETSDT for the versioned attribute VA1 in item instance ITM1 in
23 repository entity ENT1.

24
25 Control flow exits box 1102 from boxes 1305, 1309, 1313 and 1319 for the examples,
26 respectively.

27
28 It has been noted that the repository could be also be used to store information such as value
29 added functions or customer's business documents. These objects require some or all of the
30 capabilities of repository entities with item instances and versioned attributes. It is possible
31 to support the storage of such objects with repository and ETSDT's exactly as described
32 herein. An alternate embodiment involves the use of a simplified data structure for these
33 objects, encompassing storage of the object, properties to help locate it in repository, and a

1 single ETSDT with sourcing information to manage entitlement to the object. Handling the
2 addition of such an object to the store and annotating it requires some simplification and
3 omission of steps from the control flow of Fig. 9. Such modifications will be obvious to
4 practitioners of the art, after reading the material herein.

5
6 Fig. 10 expands box 1103 introduced in Figs. 7A and 7B and labeled “maintaining source
7 based entitlement information,” providing a more detailed control flow for an advantageous
8 embodiment of this box.

9
10 Control enters box 1103 whenever new source-based entitlement information arrives at the
11 repository as an input. The received entitlement information update is passed in to the flow of
12 this figure as an input parameter. Box 1401 represents receipt of the updated entitlement
13 information. Decision element 1402 is the step of determining the type of supplied
14 entitlement information update. Three types of updated entitlement information are described:
15 updated information is provided on a sourcing, on a requester or on a grant from a source to a
16 requester.

17
18 Box 1403 represents entitlement information describing a new source or source process.
19 Each source provides information on repository entities to the repository and grants particular
20 identified requesters entitlement to the provided values. In the context of a repository
21 including information on financial instruments, examples of a source are Vendor A or Vendor
22 B. Each source makes their own contractual arrangements with external entities to provide
23 raw data for a service fee. A repository that enhances and stores this information from
24 multiple sources and delivers it to multiple tenant organizations in response to requests has to
25 be able to demonstrate to each data source provider that no information has been passed to a
26 requester not entitled to receive it.

27
28 Decision element 1406 represents the separation of new sourcing information into two types:
29 value sources and process sources. Box 1407 represents processing of value sources; box
30 1409 represents processing of process sources. The previously provided source examples of
31 Vendor A and Vendor B represent examples of value sources. Value sources deliver
32 particular data services, in the form of source datasets, such as a stream of information on
33 bonds or a stream of information on corporate hierarchy, in a manner that the specific values

1 provided, and any values derived from them through the application of single-source dataset
2 based validation processes, can be accessed only by requesters who have explicitly contracted
3 with the source to receive them. Process sources represent value enhancement processes
4 typically provided as a data quality assurance and enhancement process associated with the
5 repository. Value enhancement processes are a type of an item instance process. Examples
6 include validation and cleansing of a single source dataset in isolation and a comparison
7 process using multiple source datasets providing alternate values for the same referred entity
8 to select the most reliable value. Requesters need to be entitled to an item instance process as
9 well as the attribute values used in the application of the item instance process in order to be
10 entitled to receive values generated by applying that process to those source values.. Boxes
11 1408 and 1410 represent the creation and maintenance of information uniquely identifying
12 both value and process sources, respectively, as part of the entitlement information
13 represented in data element 1418.

14
15 In addition to uniquely identifying and characterizing all sources (both process and value)
16 that may grant entitlement, the information represented by data element 1418 also identifies
17 and characterizes all requesters that receive entitlements. In an advantageous implementation
18 of a reference data utility using this repository method, the entitlement information
19 represented by data element 1418 is saved in the entitlement repository, data element 53 in
20 Fig. 1B.

21
22 Box 1405 represents entitlement information describing a new requester. Information
23 characterizing requesters is maintained so that all entitlement grants are well formed,
24 resulting in well-defined target requesters that can be authenticated. Decision element 1411
25 represents the separation of new requestor information into two types of requester: tenant
26 requester (clients) and other requesters. Box 1412 represents processing of tenant requesters,
27 which are customers of the repository. Box 1413 represents processing of other requesters,
28 which include personnel associated with the repository who provide repository maintenance
29 or customer service and, in a financial context, individuals or entities associated with audit
30 functions on behalf of exchanges, data providers, and legal or compliance review. Box 1414
31 represents maintenance of information on all such requesters (including the authentication
32 procedure used to validate that specific requests are initiated on behalf of repository
33 requesters) and ensures that this information is included in the entitlement information

1 represented by data element 1418. The information maintained on tenant and other
2 requesters and the methods used to authenticate them may differ or may be similar.

3
4 Block 1404 represents processing of an entitlement from a specific granter to an identified
5 grantee. Box 1415 represents location of the granting source within the information already
6 stored in the sourcing list represented by data element 1418. The entitlement granter may be
7 a value source, a source dataset or an item instance process. Box 1416 represented
8 identification of the requester requiring entitlement, the grantee, in the list of valid requesters.
9 Box 1417 represents the creation of the new or updated grant of entitlement (an update may
10 supplement or revoke previous entitlements) to this requester from this source for inclusion in
11 the entitlement information represented by data element 1418. As noted previously this
12 entitlement information could be stored in the repository or separately.

13
14 The entitlement information represented by data element 1418 enables enforcement of
15 current entitlements during request processing. A stream of source and requester definitions
16 and grants issued occurs, each generating separate flows at a different points in time through
17 the logic described in Fig. 10.

18
19 Fig. 11A details the overall process employed by the repository to respond to requests for
20 information based on requester preferences. Box 1104, introduced in Figs. 7A and 7B,
21 represents the overall high level flow of the process. Box 1501 represents receipt of the
22 request for information, and interpretation of the request to extract the request specification.
23 The request comes from any requester; that is any party or process acting on behalf of a
24 customer or tenant, or an agent of any data management utility or system in the context of
25 which the repository is being used.

26
27 Box 1502 represents the actions taken by the repository to locate the requested information
28 elements.

29
30 Box 1503 represents the application of entitlements, thereby limiting the set of return values
31 to those to which the requester is entitled. This is done on the basis of sourcing, which is
32 possible because information elements in the repository are annotated with sourcing
33 information as described previously. Because of this feature of the invention, the action

1 represented by box 1503 becomes largely a matter of comparing the sources and processes to
2 which the requester is entitled to the sources and processes which contributed to the requested
3 information (see Fig. 11B for some of the finer details of this process). This can be
4 contrasted with conventional systems in which entitlements typically only deal with the
5 ability of users to execute particular functions, rather than access data from particular
6 sources.

7
8 Box 1504 represents the final step of returning the resulting dataset to the requester. As
9 shown by dashed arrow 1113, it is this step which generates the response to the retrieval
10 request initially introduced as an output of the overall method 1100 in Figs 7A and 7B and
11 logs as appropriate.

12
13 In Fig. 11B, box 1501, which represents receiving the request and extracting the request
14 specification, is further decomposed into boxes 1505, 1506, and 1507. The request
15 specification received by the repository includes an arbitrary number of parameters, but at the
16 minimum, it includes the following:

17 - identification of the requester (represented by box 1505)
18 - a predicate governing selection of the information elements to be returned (represented by
19 box 1506). The selection predicate can use implementation dependent languages (such as
20 SQL) to specify which information elements the requester is interested in, and includes
21 parameters that are typically expressed by means such as interest lists, temporal restrictions,
22 conditional selection, etc.

23
24 - an ordered list or other prioritization structure specifying the requester's preference of
25 sources if multiple information elements from separate sources are available that satisfy the
26 selection predicate in the previous step. This is referred to as a sourcing preference
27 (represented by box 1507). Sourcing preference is a very important aspect of this invention
28 because it is an advantageous piece of information used to navigate a repository in which data
29 from multiple sources and belonging to multiple clients is located. The sourcing preference
30 of the requester is used in conjunction with entitlements and evolutionarily tracked source
31 data tags of information elements to ensure that requesters get only the information to which
32 they are entitled. (The entitlement enforcement aspect of this process is described in more
33 detail in Fig. 11B; also see the description of box 1503 above). It is also important to realize

1 that some sourcing preferences may have a complex multi-level structure and exist at
2 multiple information levels. For example, when creating a sourcing preference in the context
3 of financial information, it reflects the following complex preference (sample): “for European
4 stocks, the preference is: first, single-source cleansed Vendor A; if not available then single-
5 source cleansed Vendor B; if not available then normalized-only Vendor C. For US bonds,
6 the preference is: first, normalized-only Vendor A; if not available then single-source
7 cleansed Vendor C, except where the bond is classified as corporate bond: in this case, first,
8 single-source cleansed vendor C, then cleansed Vendor B. For all other bonds, the preference
9 is for single-source cleansed values from all three of Vendor A, Vendor B and Vendor C.
10 Finally, for US stocks, the preference is for values generated by a cross-source comparison
11 and selection process X”. In this example, the sourcing preference touches upon multiple
12 information levels (repository entities, item instances, attributes and metadata) and potential
13 sourcing choices, and requires multiple levels of processing to satisfy.

14
15 An example of further elaborated flow for getting the information selection predicate is
16 shown in Fig. 11C. The selection predicate part of the request specification can refer to any
17 level of information within the repository and, as such, effectively includes predicates
18 referring to any available information item, namely repository entity (represented by box
19 1509), item instance (represented by box 1510), and any attribute values (represented by box
20 1511). Once executed, the selection predicate yields zero or more information elements.

21
22 The main task of the process represented by Box 1501 in Figure 11B is to parse, validate and
23 extract the above items from the request received. The specifics of the process required to
24 parse out this information are well understood by practitioners of the art and are not the
25 subject of this invention.

26
27 In Fig. 11D, box 1502 is further decomposed into boxes 1512, 1513, 1514, 1515, and 1508
28 which show an example flow, in greater detail, of steps taken by the repository to locate the
29 information elements matching the request specification extracted above. This process is
30 aligned with the request specification aspects described in relation to box 1501. As
31 explained, the two advantageous aspects of the request specification, the selection predicate
32 and the sourcing preference, are frequently used to express quite complex concepts. To
33 satisfy the request, the repository first performs information selection at all levels as needed,

1 namely at the repository entity level, item instance level, versioned attribute and attribute
2 value level. It is possible that metadata associated with these information elements is also
3 selected. These activities are represented by boxes 1512, 1513, 1514, and 1515, respectively.
4 This process forms a return dataset, to which the requester's sourcing preference is then
5 applied, usually narrowing the dataset (represented by box 1508). This is done by comparing
6 the sources specified in the sourcing preference to the sourcing information recorded in the
7 repository for each information item. It is possible that some elements of the sourcing
8 preference cannot be satisfied (for example, no information from preferred data sources was
9 found); in this case the repository will need to include a special record reflecting this in the
10 return dataset, or use other means of notifying the requester. In an implementation of the
11 repository in the context, for example, of a multi-tenant reference data repository, multiple
12 optimization options are available to make the process of locating information elements more
13 efficient. These include controlled, data-driven methods of forming allowed requests, limits
14 or minimum requirements on the number of preferred sourcing choices, table views, various
15 repository indexing techniques, etc. However, at its functional core, any such
16 implementation remains consistent with the described steps.

17
18 In Fig. 11D, selection of information is represented by box 1502. The selected information
19 elements are then filtered through entitlements box 1503. In an alternate embodiment,
20 entitlements 1503 could occur before or as part of 1502. When this is done all of the actions
21 within box 1502, specifically 1512, 1513, 1514, 1515, and 1508 are subject to entitlements.
22 They each return a response based on the entitlements of the requester.

23
24 Fig. 11E provides additional detail about the activities represented by box 1503 from Fig.
25 11A, namely, enforcing entitlements as part of the process of responding to a request. The
26 multi-source, multi-tenant nature of the repository makes processing entitlement information
27 a more complicated task than a simple filtering scheme that might be employed in single-
28 tenant data management applications. Specifically, it is insufficient to enforce entitlements at
29 a single point (for example, at the lowest data structure level – the attribute) because a multi-
30 source multi-tenant data repository supports storing item instances generated by cross-source
31 processes (a type of item instance process) which may themselves require entitlement.
32 Further, it is possible to be entitled to a process, yet not be entitled to all values that this
33 process generates, which is why a multi-level entitlement check takes place. For instance,

1 continuing with the example of a financial instrument reference data repository, a reference
2 data utility in which the repository exists may offer, as an additional service, a multi-source
3 item instance process P that produces composite records based on multiple sources according
4 to some algorithm. Tenant A of the repository subscribes to this service. However, based on
5 the rules driving the service, the composite records it generates sometimes include
6 information from a data source to which tenant A is not entitled. In these cases, such results
7 are not returned to tenant A, even though tenant A is subscribed to the service. The two-
8 level source check (process level and attribute value level) is required to detect and properly
9 handle such situations. Optimizations include designating separate terms like “simple
10 source” and “complex source” to help differentiate at runtime between item instance
11 processes that require one-level entitlement checking vs. two-level entitlement checking. At
12 its functional core, the entitlement checking process is aware of and accommodates both
13 possibilities.

14
15 In Fig. 11E, the entitlement process is represented by box 1503 starting at the repository
16 entity level (i.e. the desired repository entity has already been located). Box 1516 represents
17 the retrieval of the requester’s entitlement to item instance processes of the current repository
18 entity using the entitlement information represented by data element 1418 as shown in Fig.
19 10. This entitlement information, and the steps required to create it, were described in Fig. 10.
20 Box 1517 represents a check based on this entitlement information to determine whether this
21 requester is entitled to access the selected item instances (recall that each item instance is
22 associated with an item instance process). It is at this level that information about the item
23 instance process that generated the given item instance is stored. Additional information
24 stored in the ETSDT for the item instance may also need to be used, as represented by the
25 dashed line connecting box 1517 with data element 1220. Decision box 1518 represents a
26 flow checkpoint; if the check represented by box 1517 fails, the requester is not entitled to
27 access this item instance; if the check succeeds, further checking at the attribute level occurs.
28 In the event of a successful outcome at decision element 1518, box 5119 represents retrieval
29 of the requester’s entitlement to specific sources from the entitlement information represented
30 by data element 1418. In an alternative implementation this step is combined with activities
31 represented by box 1516. Box 1520 represents the actual entitlement check at the attribute
32 level. This check utilizes sourcing information from a versioned attribute ETSDT (data
33 element 1227) to ensure that only entitled sources have been used to produce the desired

1 value. If the check passes (at the decision point represented by decision box 1521), the
2 attributes and the enclosing item instance are entitled and are eligible to be returned to the
3 requester. Otherwise, based on the nature of the item instance process, either the specific
4 versioned attributes or the entire item instance is removed from the return set (represented by
5 box 1522). This process proceeds across all selected item instances and selected attributes to
6 produce a filtered dataset that is returned to the requester. This concludes the description of
7 the flow diagrams pertaining to the repository aspect of the invention. If the test in block
8 1518 fails, then no entitle item instance is available so control flows out of block 1503.

10 C. Description of Data Cleansing and Value Enhancement

12 This section describes a method and organization for performing scalable data cleansing and
13 value enhancement of arriving reference information in which both single data source
14 enhancement processing and multiple data source comparison and enhancement processing
15 are supported while the method still maintains full knowledge of all sources used in deriving
16 reference data elements. In the context of a reference data utility, this method can provide the
17 data acquisition and quality enhancement processing shown as box 19 in Fig. 1A.

19 Figs. 12 A and 12B when taken together show a complete high level control flow for the Data
20 Cleansing and Value Enhancement method (DCVE). Fig. 12A shows the single-source data
21 cleansing portion of the DCVE. Figure 12B shows the multisource data processing.

23 In Fig 12A the vendor sources of data are represented by ellipses 2101, 2102, 2103. Multiple
24 sources of data are concurrently processed by the DCVE. In Fig. 12A each source,
25 represented by ellipses 2101, 2102, and 2103, is providing a dataset on reference data topic
26 T1. In the context of a reference data utility, this corresponds to the T1 introduced as box 22
27 in Fig.1A. Arrows 2132, 2133, and 2134, represent control transfers when single source
28 DCVE processing is complete and multiple source DCVE processing in Fig. 12 can be
29 initiated. Fig. 12A describes at a high level how source attributes are processed for this
30 dataset. Source items are processed in a similar manner. More detail on source and attribute
31 processing is given in Fig. 14.

1 In general, data is received and processed for multiple topics in this component. Topics are
2 properties that enable hierarchical organization within the repository. Examples of separate
3 reference topics in a financial reference data repository include:

- 4 - reference data on financial instruments;
- 5 - corporate hierarchy and counterparty information; and
- 6 - corporate action event notification.

7
8 The DCVE processing of separate topics is independent. However, the same source
9 descriptions are used for any common concepts and, in the advantageous embodiment, the
10 received qualified reference data values are stored into the same repository. The source
11 description contains information describing structure, contents and constraints on data within
12 datasets provided by a particular source.

13
14 Fig. 12A shows the DCVE processing for three data sources supplying reference data values,
15 source S1, source S2 and source S3 represented as ellipses 2101, 2102, and 2103,
16 respectively. There can be any number of sources of data values on a specific topic divided
17 between licensed vendors, free public sources and qualified on-demand sources. In our
18 description of this figure we are assuming that the sources are supplying data for the same
19 topic. This assumption allows us to illustrate cross source processing in Fig. 12B. However,
20 the DCVE processes data from multiple sources on different topics concurrently. The DCVE
21 processes as many sources and topics as are available and is not limited to processing three
22 concurrently. DCVE processing treats each source as an independent dataset of reference
23 data values. Elements 2105, 2111, 2120, 2129, 2114, and 2123 deal with source S1 values;
24 elements 2106, 2112, 2121, 2130, 2115, and 2124 deal with source S2 values, and elements
25 2107, 2113, 2122, 2131, 2116, and 2123 deal with source S3 values. The repository is
26 represented by elements 2108, 2109 and 2110. We represent this as separate storage for each
27 stream to show that the intermediate processing results during the DCVE processing are
28 managed independently for each stream. In an advantageous implementation of a reference
29 data utility using this DCVE method for input processing, this storage would be provided
30 within a single utility repository as shown as element 20 in Fig. 1A. Separate DCVE
31 processing of each source dataset enables the recording of the source of each processed value.

32

DCVE processing for source S1 values is described in greater detail; the corresponding processing of the other sources is similar. DCVE processing of a single source proceeds in steps:

- attribute and item validation and creation of ETSDT, represented by box 2105 and ellipse 2129 for source S1;
- attribute and item normalization, represented by box 2111 and ellipse 2114 for source S1;
- and
- source-specific attribute and item value cleansing, represented by box 2120 and ellipse 2123 for source S1.

The modified attribute and item values are stored in the repository. All of the events and sources used to create the modified values are recorded as ETSDT annotations also contained in the repository. The repository is represented by element 2108. These steps are sometimes followed by a step that applies one or more processes of cross-source attribute value comparison, potentially using data from a variety of sources providing data on this topic. This is illustrated in Fig. 12B described below.

Box 2105 represents the first step inside the DCVE component; receiving and processing datasets arriving from source S1. This step handles the receive protocol and getting the dataset from source S1 into the repository. Attribute validation processing usually includes:

- authentication of source, acknowledge, protocol and format handling;
- assignment of unique identifiers and/or timestamps to input records;
- verification that the source attribute values conform to the source description; and
- manual validation for any elements of the dataset that cannot be automatically validated.

After receiving the dataset and validating it for acceptance into the DCVE component, the validated attributes are stored in the repository and events arising from validation of the attributes from source S1 are logged, as represented by arrow 2181, into the ETSDT(s), which are also stored in the repository. The repository is represented by box 2108. This logging is done by recording the results of validation, actions taken during validation, and the completion of the attribute validation as ETSDT annotations.

It is possible that anomalies are present in the received dataset that cannot be validated automatically. When this occurs, those parts of the dataset are passed to manual validation,

1 represented by ellipse 2129, where a human with business knowledge corrects the errors if
2 possible. After manual validation, the validated attributes are stored in the repository and the
3 events that arise during manual validation from source S1 are logged, as represented by arrow
4 2151, as ETSDT annotations.

5
6 Box 2111 represents the automated attribute normalization processing of the arriving data
7 from source S1. This step deals with the issue that particular reference data attributes may be
8 referred to with different attribute names by different dataset sources. Furthermore, particular
9 attribute values for the reference data item may be represented in a different way in different
10 sources. Dashed arrow 2171 shows validated data from the preceding manual or automatic
11 validation step being made available as input to automatic normalization 2111.

12
13 The target description contains information describing the structure, contents and constraints
14 on repository entity information, including item instances, versioned attributes and attributes
15 as they are stored in the repository. Received attributes for a reference data item are
16 translated into a standard representation. Attribute normalization processing usually includes
17 mapping the source attribute from the source description to a target attribute based on the
18 target description. This process looks up the reference data attribute supplied by source S1 in
19 a source description so that the standard attribute name is matched. Looking up and
20 translating the attributes is done automatically by applying a set of lookup and automated rule
21 steps for efficiency reasons. This includes transforming source attribute values to target
22 attribute values. The normalized attribute names and values are stored in the repository. The
23 events and sources used to create the normalized attribute names and values are recorded as
24 ETSDT annotations, as represented by arrow 2182.

25
26 Sometimes attribute name and value lookup fails or other anomalies are detected during the
27 automated attribute normalization step. For each exception case the problem reference data is
28 forwarded to the manual attribute normalization processing step represented as ellipse 2114.
29 In this step, a human with business knowledge and skilled in the subject topic decides
30 whether to accept or how to modify the anomalous value. For example, the human decides
31 whether a financial instrument entity whose name was not in the source description is a
32 newly created type of financial instrument which has not been seen before and needs to be
33 added to the source description or whether the name is a misspelling or other data input error

1 of an existing named instrument. The normalized attribute names and values are stored in the
2 repository. The events and sources used to create the normalized attribute names and values
3 are recorded as ETSDT annotations and stored in the repository, as represented by arrow
4 2152.

5
6 After a received reference data attribute is normalized, either by automatic processing or after
7 inspection and possible manual correction, the normalized attributes are stored in the
8 repository and the events used to normalize the attributes from source S1 are logged, as
9 represented by arrows 2182 and 2152 respectively, into the ETSDT(s). This logging is done
10 by recording the results of normalization, actions taken during normalization, and the
11 completion of the attribute normalization as ETSDT annotations.

12
13 After attribute normalization is completed, arriving reference data from source S1 goes
14 through a source-specific item cleansing process as represented by boxes 2120 and 2123.
15 The purpose of source-specific item cleansing is to verify the correctness of the data content
16 through the application of business rules, without reference to any other source.

17
18 The first step is an automatic cleansing phase, which is represented by box 2120. Dashed
19 arrow 2172 shows normalized data saved in the previous normalization step being made
20 available as input to automatic cleansing. In step 2120, automated cleansing checks for
21 missing data, garbled data, data values out of expected range (range tolerance), data which
22 has changed by some unreasonable shift from the previously known value (rate of change),
23 how well-formed the data is, consistency with the target item instance (described by the
24 target description), compatibility with well known referred entities of similar target
25 description, sensitivity to recent news, and other programmable source attribute value checks.
26 These checks are based on the information contained in the source and target descriptions.
27 Again, for efficiency reasons, in order to filter through the bulk of arriving data which will be
28 required to pass all of these tests, it is advantageous for the initial cleansing phase to be
29 automated. The cleansed attributes are stored in the repository and the events and sources
30 used to create the cleansed attributes are recorded as ETSDT tag annotations and also stored
31 in the repository, as represented by arrow 2183.

1 Some items fail the automatic cleansing checks represented by box 2120 and are separated
2 out as exceptions and passed to manual cleansing represented as ellipse 2123. At this point, a
3 human with business knowledge and skilled in the subject topic reviews the excepted items
4 and decides whether to accept, reject, or to correct the arriving anomalous normalized value.
5 This source specific item cleansing is still done only with reference to data arriving from
6 source S1. Freely distributed public information is used to improve, cleanse or augment data,
7 but no other vended licensed data is used. This constraint is necessary in order to avoid
8 contaminating data ownership and right of access to the other sources. The use of freely
9 available information can also be logged. The cleansed attributes are stored in the repository
10 and the events and sources used to create the cleansed attributes are recorded as ETSDT tag
11 annotations, also stored in the repository, as represented by arrow 2153.

12
13 After a normalized attribute is cleansed, either by automatic processing or after inspection
14 and possible manual correction, the cleansed normalized attribute is stored in the repository
15 and the events used to create the cleansed normalized attribute from source S1 are logged to
16 the repository, as represented by arrows 2183 and 2153 respectively, in the ETSDT(s). This
17 logging is done by recording the results of cleansing, the actions taken during cleansing, and
18 the completion of the cleansing as ETSDT annotations.

19
20 In an alternate embodiment cleansing of the arriving dataset from a source is performed first
21 and normalization afterwards. The advantage of the ordering shown above is that the valuable
22 human resource used to inspect and manually cleanse arriving data is more freely assignable
23 from one source to another if they are familiar with reviewing already normalized values.

24
25 Error detection usually results in manual steps: manual normalization (ellipse 2114), manual
26 validation (ellipse 2129), and manual cleansing (ellipse 2123); and/or causes the feedback or
27 problem reporting, represented by arrows 2135, 2150, and 2176, to the dataset source (ellipse
28 2101). Typically, if an error or problem is found or thought likely in a reference data value
29 received from source S1, the data provider is notified and asked to confirm or correct the
30 provided value.

31
32 This style of feedback between DCVE processing and sources is best handled by making
33 further use of the ETSDT. Values which have passed through the DCVE processing without

1 issue are tagged as normal. Other values are passed on for potential use but tagged as
2 'questionable' or 'awaiting confirmation'. Values tagged this way are typically used by those
3 repository tenants who need to receive updated values in real-time despite the probability of
4 error. When a source provides an updated or confirmed value in response to notification that
5 a previous value received from them was tagged 'questionable,' the updated value is
6 processed with a corresponding normal tag.

7
8 After single source validation, normalization, and cleansing is complete, the cleansed and
9 enhanced data is made available for one or more multiple source DCVE processes. Arrow
10 2132 shows the flow of control conveying single source DCVE processed data from source
11 S1 to a multiple source DCVE process in Fig. 12B. Similarly arrows 2133 and 2134 represent
12 single source DCVE processed data from sources S2 and S3 respectively being made
13 available to the same example multiple source data cleansing process in Fig. 12B. The
14 single source DCVE processing of data from sources S2 and S3 were handled by independent
15 parallel processing similar of structure to the method we have describe in detailed as applied
16 to the single source DCVE processing for the data from source A.

17
18 In the example shown here with Figs. 12A and 12B we show three sources each being
19 cleansed individually then the results being used as input to a single multiple source DCVE
20 process. The method can be generalized from this description and can be applied to
21 individual single source cleansing of any number of sources, followed by a stage of
22 delivering the results from any one single source DCVE process to any number of multiple
23 source DCVE processes.

24
25 Automated workflow management techniques may be used to facilitate coordination and
26 management of the manual steps 2129, 2114, 2123, 2130, 2115, 2124, 2131, 2116, and 2125.
27 There are a number of alternative implementations such as semaphores or loosely coupled
28 distributed processes. Those skilled in the art know how to coordinate asynchronous
29 processes. The exact mechanism used to coordinate the individual steps of the described
30 flows is not important to this process. There are many techniques known to the practitioners
31 of the art which can be used for these purposes.

Fig. 12B illustrates the cross-source cleansing value enhancement portion of the Data Cleansing and Value Enhancement process (DCVE) that is applied after source-specific item cleansing has been completed. The DCVE process may apply one or more cross-source item comparisons and/or cross-source item cleansing processes. One example of such a cross-source process provides the selection of a recommended value for a normalized attribute across all source datasets. This example is used for illustration of the concepts of this figure. The basic components of this process are represented by box 2138 and ellipse 2170.

Arrows 2132, 2133 and 2134 from Figure 12A to the automatic select and enhance step represented by box 2138 represent transfer of control to the multiple source DCVE processing of Fig. 12B when new single source DCVE processed data becomes available from sources S1, S2 or S3. The method of synchronization is not important for the invention. In general as soon as new data from any of the input sources is available this can be compared with previously received values from this and other sources and a level of multisource DCVE processing can occur. In other cases it may be efficient to batch the multisource processing following some fixed schedule or when a full set of single source cleansed data is available for a particular reference entity from all expected sources. The processing of box 2138 uses the separately normalized and cleansed values from some subset of source datasets for this topic, applying automated business rules to select a preferred or recommended value for this reference data item. Arrows 2191, 2192 and 2193 represent retrieval of these values from the repository where they were stored in as saved data during the single source processing of Fig. 12A represented by store elements 2108, 2109, 2110.

The resulting recommended cross-source compared and cleansed values are then stored in the repository, as represented by arrow 2194. The events and sources used during the process of cross-source cleansing, as well as the completion of the cross-source cleansing process are recorded as ETSDT annotations, which is reflected by arrow 2194 as well. ETSDTs are also stored in the repository represented by element 2140. As noted above this element shows that the results of a particular multiple source DCVE process are saved to make them accessible to subsequent requesters entitled to values from this value creation process. In the context of a reference data utility, store element 2140, along with store elements 2108, 2109, 2110 would share a common store for entitlement managed entity data as was represented as element 50 in Fig 1B as part of the utility repository 20.

1 When the automated process cannot arrive at desired results, manual intervention is
2 employed, as shown by element 2170. The resulting recommended cross-source compared
3 and cleansed values are then logged, as represented by arrow 2175, in the ETSDT. The
4 events arising from this manual process are similarly logged as ETSDT annotations in the
5 repository 2140. This logging is also shown by element 2175.

6
7 All source datasets received, validated, normalized, cleansed and prepared as target datasets,
8 along with any attribute values enhanced through cross-source comparison and/or cleansing
9 processes, are stored separately in the ETSDT repository. Each of these datasets of reference
10 data values has clearly understood sourcing. Multiple cross-source dataset processes in the
11 DCVE result in datasets in an ETSDT tagged with all the referenced sources. All cross-
12 source processes that produce datasets store the actions undertaken in ETSDTs with all
13 referenced sources logged. The ETSDTs are stored in the repository represented by element
14 2140. In an alternate embodiment it is possible to use a different number of ETSDTs as
15 appropriate.

16
17 Automated workflow management techniques facilitate coordination and management of the
18 control transfers 2132, 2133, 2134 and processing steps 2138, and 2170. There are a number
19 of alternative implementations such as semaphores or loosely coupled distributed processes.
20 Those skilled in the art know how to coordinate processes.

21
22 The detailed flow for DCVE processing for a single topic is described herein. This processing
23 is repeatable for each reference data topic, with the understanding that:

- 24 - there may be qualitative differences in that some topics are driven almost entirely by
- 25 licensed feeds with atomic instrument data; and
- 26 - topics such as corporate and counterparty hierarchies may have more coupled records and
- 27 require more activist data gathering.

28
29 Despite these qualitative differences in emphasis, the pattern and structure of data,
30 acquisition, quality assurance and enhancement are essentially the same across topics. The
31 net effect of the data acquisition, cleansing and enhancement process is to provide a
32 “production line” approach for receiving and engineering a high level of quality of reference
33 data while completely preserving auditable and transparent ownership of the data.

Fig. 13 provides a high level overview of the processes of validation, normalization, single-source cleansing and multi-source processing. The term “multi-source processing” rather than “multi-source cleansing” is used to denote that multi-source processes vary greatly in nature and encompass not only basic quality assurance of data, but also select between incompatible values, generate new values based on several sources, or any other programmable process which references multiple sources of data. Fig. 13 particularly stresses the interactions with ETSDTs of respective information elements at the various steps of the described processes.

The first column, headed by box 2200, describes the validation process. This corresponds to the processing of steps 2105, 2106, 2107, for an automated version, and 2129, 2130 and 2131 for a manual version in Fig. 12A. Validation is typically the first process applied to an arriving dataset, and its function is to perform basic structure and content validation. The first step is to extract source items from the dataset, represented by box 2201. This is typically done based on the source dataset description supplied by the data provider, which normally details headers, record structures or delimiters and similar information. Once source items are extracted, a fully tracked history for each source item begins. Box 2202 represents the creation or update of an ETSDT for each source item to record the events of the source item’s history. One of the first pieces of information recorded in the ETSDT is the source of the item, represented by box 2203. Because later on the information collected in items may no longer be grouped by source, it’s very desirable to preserve source information at the lowest level available. Once this is done, validation rules are applied to the source item, as represented by box 2204. The rules are typically created based on source description information and exist at source item level and attribute level. In some embodiments there may be no rules which apply to a source item. Box 2205 represents annotation of the ETSDT to reflect the application of source item level rules. The information stored includes which rule was applied and the outcome of applying the rule (e.g. pass/fail). If a correction was applied, that is recorded as well. When corrections are applied (at any level), the original record is not overwritten, but kept as a previous version, with the ETSDT serving as the history detailing such information as when, why, and during which processes corrections were made. If the correction has a specific source (for instance, if a correction was applied manually by an employee who used an original business document as a source), this is recorded in the ETSDT as well.

1 Once source item level validation rules are applied, processing moves to the attribute level.
2 Similar to the process applied to extract source items from the source dataset, box 2206
3 represents extraction of attributes from each source item. Following this, an ETSDT is
4 created for each attribute and the original source of the attribute is recorded in the ETSDT,
5 actions represented by boxes 2207 and 2208, respectively. Attribute level rules are applied
6 (box 2209) and all the resulting events and sources associated with rule application are
7 recorded in the ETSDT (box 2210).

8
9 The process, 2200 – 2211, is repeated for all source items and attributes.

10
11 Box 2211 represents a notation to the ETSDT indicating that a source item processed in the
12 above manner has gone through validation. Validation is an example of an item instance
13 process in which information in a dataset has been affected in some manner by the repository.
14 Recording the item instance processes which have been applied to a source item is a desirable
15 operation as this is essential to maintaining an auditable history of the data.

16
17 The second column of Fig. 13, headed by box 2212, describes the process of normalization,
18 which typically follows validation. This corresponds to the processing of blocks 2111, 2112,
19 2113, for an automated version and 2114, 2115 and 2116 for a manual version in Fig 12A.
20 At this point, the source items have already been extracted from the original source dataset,
21 and are selected one by one to be normalized, a process represented by box 2213. Each
22 source item (box 2214) is normalized in the manner employed by standard extract-transform-
23 load (ETL) processes – structure modifications, code lookups, application of standards, and
24 similar processes. Changes made during this process can be at the source item level (e.g.
25 structural) and/or attribute level (e.g. date format), and are recorded as annotations in the
26 ETSDT at the source item level, as represented by box 2215, or attribute level, as represented
27 by box 2216. As with the validation process, the original version of the item is retained. Box
28 2217 represents annotation of the item ETSDT at completion of the normalization process,
29 indicating that the item has undergone the process of normalization (Box 2217).

30
31 Single-source cleansing, headed by box 2218, is shown in the third column. This corresponds
32 to the processing of boxes 2120, 2121 and 2122 in an automated version and boxes 2123,
33 2124 and 2125 in a manual version. Box 2219 represents the first step of selecting an item

1 for cleansing. As not all source items need to be cleansed, performance of this step is based
2 on preliminary flagging, a random sampling algorithm or some other algorithm as necessary.
3 During cleansing there are rules that apply at source item level (e.g. problems with
4 correlation between different attributes of an item) or at an attribute level (e.g. a price is too
5 far beyond a certain threshold). As box 2220 represents, source item level rules are applied
6 first. Then, as represented by box 2221, events generated during the application of these rules
7 are recorded in the item level ETSDT as before. Attributes are selected and rules are applied
8 at attribute level, as represented by boxes 2222 and 2223, respectively. The events are
9 recorded, represented by box 2224, in the attribute level ETSDT. As with the other
10 processes, the final box 2225 represents annotation of the source item level ETSDT at
11 completion of the process to show that the item has gone through the single source cleansing
12 item instance process.

13
14 The final column of Fig. 13 shows cross-source processing headed by box 2226. This
15 corresponds to the processing of box 2138 in automated form and 2170 in manual form in
16 Figure 12B. Cross-source processing is especially interesting because items from multiple
17 sources which refer to the same real-life entity (referred entity) are involved. This requires
18 especially careful recording of the item and attribute sources.

19
20 Cross-source processing begins with selection of all of the source items that contain
21 information describing the same referred entity. This is represented by box 2227. For
22 example, if IBM common stock is the referred entity, the item from source A, source B and
23 source C, representing IBM common stock as provided by these different sources, would be
24 selected. Next, box 2228 represents application of the rules to the source items and/or
25 attributes of the items. Because a rather large number of possible cross-source processes
26 exist, further detail is not shown. However, most cross-source processes tend to fall into one
27 of the following categories:

- 28 - processes that only select the "best" or otherwise preferred or recommended item from the
- 29 alternatives provided by the different sources;
- 30 - processes that create new items based on some combination of attributes provided by the
- 31 different sources; or
- 32 - processes that modify in place the items provided by the different sources.

1 For those processes that create a new item or items, a new corresponding ETSDT is created.
2 This is represented by the decision box 2229 and box 2230. Box 2231 represents the
3 annotation of the ETSDT at the source item level with the information about the cross-source
4 processing applied to the item. At runtime, this annotation identifies exactly what kind of
5 cross-source process was applied. Box 2232 represents a decision point that distinguishes
6 handling of cross-source processes that only select preferred or recommended item from the
7 other processes. If the cross-source process was of this type, i.e. an existing item was selected
8 but no attributes were actually modified, then an annotation is made at the source item level
9 to denote which parent sources matched the selection made, as represented by box 2233. For
10 instance, if an item representing IBM common stock with price of \$95.50 was selected, it's
11 possible that more than one source participating in the cross-source process contributed the
12 same data. In this case, the annotation represented by box 2233 would include all such
13 sources. Alternatively, if the cross-source process is of one of the two other types, that is, if it
14 includes either modification of data at an attribute level or a creation of a new source item
15 altogether, then it is necessary to annotate the exact set of sources for each attribute
16 separately. In this case, box 2234 represents appropriate annotations at the attribute level for
17 each impacted attribute. Multiple sources per attribute are also possible.

18
19 The exact mechanism used to coordinate the individual steps of the described flows is not
20 important to this process. There are many techniques known to the practitioners of the art that
21 are used for these purposes.

22
23 Fig. 14 shows the processing required to perform single-source dataset validation. This
24 process was first described in Fig. 12A, box 2105 and elaborated in Fig. 13, elements 2200
25 through 2211.

26
27 During this process the original item values and original attribute values as well as all
28 modifications to those values are stored in the repository. Box 2320 represents where the
29 item ETSDT is updated and box 2321 represents where the attribute ETSDT is updated.

30
31 Commencement of validation is represented by box 2305. All of the rules applied in this step
32 are source-specific; no cross-source processing is allowed. Next, as represented by box 2307,
33 the source is validated and the dataset is received. If the source is invalid the dataset is

1 recorded and the entire dataset is sent to manual processing for source validation. Otherwise,
2 a record of the receipt of the dataset is made and the rules for validating this dataset are
3 acquired, activities represented by boxes 2309 and 2310, respectively. These rules are in a
4 file, database, or other appropriate store. Box 2312 represents extraction of the first source
5 item from the dataset. The item and its source are recorded and the ETSDT is created; boxes
6 2314 and 2316 represent these activities.

7
8 The first applicable rule is applied to this item, represented by box 2318. If the item passes
9 rule application, a decision represented by diamond 2322, then an additional query is
10 performed, as represented by diamond 2350, to search for additional rules. If an additional
11 rule is found, the rule is applied to the item, again represented by box 2318. If an item does
12 not pass rule application as represented in diamond 2322, then the error is recorded in the
13 ETSDT, represented by box 2325. After the error is recorded, the system attempts automatic
14 correction, represented by box 2330, based on the information in the applied rule or in rules
15 for correcting errors. Success or failure of the attempted correction is represented by diamond
16 2335. Box 2345 represents the action taken if the problem cannot be corrected, where the
17 item is flagged as needing correction. After item flagging, the process continues to search for
18 more rules, the same query represented by diamond 2350 as explained above. If the item is
19 automatically corrected, the correction and the rule used to make the correction are recorded
20 in the ETSDT, represented by box 2340. The process continues to search for more rules.

21
22 If the query represented by diamond 2350 returns no additional rules that apply to the item,
23 then extraction of an attribute associated with this item occurs, as represented by box 2360.
24 The attribute and its source are recorded and the ETSDT is created or updated, as represented
25 by boxes 2362 and 2364, respectively. Box 2366 represents application of the first applicable
26 rule to the attribute. If the attribute passes the rule application, a decision represented by
27 diamond 2368, then an additional query is performed, as represented by diamond 2390, to
28 search for additional rules. If an additional rule is found, the rule is applied to the item, again
29 represented by box 2366. If an attribute does not pass rule application as represented by
30 diamond 2368, the error is recorded in the ETSDT, represented by box 2370. After the error
31 is recorded, the system attempts automatic correction, represented by box 2372, based on
32 information contained in the applied rule or in rules for correcting errors. Success or failure
33 of the attempted correction is represented by diamond 2374. If the error is automatically

1 corrected, the correction and the rule used to make the correction are recorded in the ETSDT,
2 represented by box 2378. The process continues to check for more attribute rules. Box 2376
3 represents the action taken if the error is not automatically corrected, where the attribute is
4 flagged as needing correction. After item flagging, the process continues to search for more
5 rules, the same query represented by diamond 2390 as explained above.

6
7 If the query represented by diamond 2390 returns no additional rules that apply to the
8 attribute, then the process searches for additional attributes, as represented by diamond 2392.
9 If another attribute is found, it is extracted (box 2360) and the rule check for the new attribute
10 proceeds. If the query represented by diamond 3292 returns no additional attributes for the
11 item, the process searches for additional items in the dataset, a query represented by diamond
12 3294. If this query finds an additional item, then, as represented by box 2312, item and
13 attribute checking starts for the new item. If the query represented by diamond 2394 returns
14 no additional items, we check to see if any errors were found during source dataset
15 processing, as represented by diamond 2396. If no errors are found the validation process
16 terminates (block 3280). If errors are found, all of the items and attributes determined as
17 needing correction are scheduled for manual validation (or manual correction), represented
18 by box 2385, and the validation process terminates (block 2380).

19
20 The exact mechanism used to schedule manual validation and pass control to it while
21 concurrently continuing processing of the parts of the dataset that are not in error is not
22 important to this process. There are many techniques known to the practitioners of the art
23 which can be used for these purposes.

24
25 Fig. 15 shows the processing required to perform normalization of a source input stream,
26 which is represented as box 2111 in Fig. 12A. This process is elaborated in boxes 2212
27 through 2217 of Fig. 13.

28
29 During this process the original item values and original attribute values as well as all
30 modifications to those values are stored in the repository. Box 2320 represents where the
31 item ETSDT is updated and box 2321 represents where the attribute ETSDT is updated.

1 Box 2405 represents commencement of normalization. Next, as represented by box 2407, the
2 validated dataset is received. A record of the receipt of the dataset is made and the rules for
3 normalization of this dataset are acquired, as represented by boxes 2409 and 2410,
4 respectively. Because this is a single-source normalization process, all of the rules are source
5 specific and do not rely on data or information from any other source. These rules are in a file,
6 database, or other appropriate store.

7
8 The first item is extracted from the dataset, as represented by box 2412, followed by
9 application of the first rule to this item, as represented by box 2418. If the item passes the rule
10 application, as represented by decision diamond 2422, then the dataset is checked for
11 additional applicable rules, as represented by diamond 2450. If an additional rule is found, it
12 is applied to the item (box 2418). If an item does not pass rule application as represented by
13 decision diamond 2422, then the error is recorded in the ETSDT, represented by box 2425.
14 After the error is recorded, the system attempts automatic correction, represented by box
15 2430, based on the information in the applied rule or in rules for correcting errors. Success or
16 failure of the attempted correction is represented by diamond 2435. Box 2445 represents the
17 action taken if the problem cannot be corrected, where the item is flagged as needing
18 correction. After item flagging, the process continues to search for additional rules, the same
19 query represented by diamond 2450 above. If the item is automatically corrected, the
20 correction and the rule used to make the correction are recorded in the ETSDT, represented
21 by box 2440. The process continues to search for more item rules.

22
23 If the query represented by diamond 2450 returns no additional rules that apply to the item,
24 then extraction of an attribute associated with this item occurs, as represented by box 2460.
25 The first applicable rule is applied to the attribute, as represented by box 2466. If the attribute
26 passes the rule application, a decision represented by diamond 2468, the dataset is checked
27 for more attribute rules, as represented by diamond 2490. If an additional rule is found, it is
28 applied to the attribute (box 2466). If an attribute does not pass the rule application
29 represented by diamond 2468, then the error is recorded in the ETSDT, represented by box
30 2470. Box 2472 represents attempted automatic correction of the error based on information
31 contained in the applied rule or in rules for correcting errors. Success or failure of the
32 attempted correction is represented by diamond 2474. If the error is successfully corrected
33 then the rule that corrected the error along with the correction is recorded in the ETSDT, as

1 represented by box 2478. The process continues to check for more applicable attribute rules.
2 If the error is not automatically corrected, the attribute is flagged as needing correction, as
3 represented by box 2476. After item flagging, the process continues to check for more
4 applicable attribute rules.

5
6 If no additional rules are found in decision diamond 2490, the item is checked for additional
7 attributes, as represented by decision diamond 2492. If another attribute is found, it is
8 extracted and the rule check (2460) for the new attribute proceeds. If no additional attributes
9 are found, the dataset is checked for additional items, as represented by diamond 2494. If an
10 additional item is found, it is extracted, box 2412, from the dataset and item and attribute
11 checking starts. If no additional items are found, the process checks to see if any errors were
12 found during source data processing, as represented by diamond 2496. If no errors were
13 found, the normalization process terminates (box 2480). If any errors are found, all of the
14 items and attributes determined as needing correction are scheduled for manual normalization
15 (or manual correction), represented by box 2485, and the automatic normalization terminates
16 (box 2480).

17
18 The exact mechanism used to schedule manual normalization and pass control to it while
19 concurrently continuing processing of the parts of the dataset that are not in error is not
20 important. There are many techniques known to the art which can be used for these purposes.

21
22 Fig. 16 shows the processing required to do perform dataset cleansing, which is represented
23 as box 2120 in Fig. 12A. This process is elaborated in boxes 2218 through 2225 of Fig. 13.

24
25 During this process the original item values and original attribute values as well as all
26 modifications to those values are stored in the repository. Box 2520 represents where the
27 item ETSDT is updated and box 2521 represents where the attribute ETSDT is updated.

28
29 Box 2505 represents the commencement of cleansing. Next, box 2507 represents receipt of
30 the validated dataset. A record of the receipt of the dataset is made and the rules for
31 cleansing this dataset are acquired, as represented by boxes 2509 and 2510, respectively.
32 Because this is a single source cleansing process all of the rules are source specific to the

1 dataset and do not rely on data or information from any other source. These rules are in a file,
2 database, or other appropriate store.

3
4 The first item is extracted from the dataset and the first applicable rule is applied to this item,
5 as represented by boxes 2512 and 2518, respectively. If the item passes rule application,
6 represented by decision diamond 2522, then the dataset is checked for more applicable rules,
7 as represented by diamond 2550. If an additional rule is found, it is applied to the item in box
8 2518. If an item does not pass rule application, represented by decision diamond 2522, then
9 the error is recorded in the ETSDT, as represented by box 2525. After the error is recorded
10 the system attempts automatic correction, represented by box 2530, based on the information
11 in the rule or in rules for correcting errors. Success or failure of the attempted correction is
12 represented by diamond 2535. Box 2545 represents the action taken if the problem is not
13 corrected, where the item is flagged as needing correction. After item flagging, the process
14 continues to search for additional rules, the same query represented by diamond 2550 above.
15 If the item is automatically corrected the correction and the rule used to make the correction
16 are recorded in the ETSDT, as represented by box 2540. Then processing continues to search
17 for more applicable item rules.

18
19 If the query represented by diamond 2550 returns no additional rules that apply to the item,
20 then extraction of an attribute associated with this item occurs, as represented box 2560. The
21 first applicable rule is applied to the attribute, as represented by box 2566. If the attribute
22 passes the rule application, a decision represented by diamond 2568, the dataset is checked
23 for more applicable rules, as represented by diamond 2590. If an additional rule is found, it is
24 applied to the attribute (box 2566). If an attribute does not pass the rule application
25 represented by diamond 2568, then the error is recorded in the ETSDT, represented by box
26 2570. Box 2572 represents automatic correction of the error based on information contained
27 in the rule or on rules for correcting errors. Success or failure of the attempted correction is
28 represented by diamond 2574. If the error is successfully corrected then the rule that
29 corrected the error along with the correction is recorded in the ETSDT, represented by box
30 2578. Then processing continues to check for additional applicable attribute rules. If the error
31 is not automatically corrected, the attribute is flagged as needing correction, as represented by
32 box 2576. After item flagging, the process continues to check for more applicable attribute
33 rules in decision diamond 2590.

1 If no additional rules are found, the item is checked for additional attributes, as represented
2 by decision diamond 2592. If another attribute is found, it is extracted in box 2560 and the
3 rule check for the new attribute proceeds. If no additional attributes are found, the dataset is
4 checked for additional items, as represented by diamond 2594. If an additional item is found,
5 it is extracted in box 2512 from the dataset and item and attribute checking starts. If no
6 additional items are found, the process checks to see if any errors were found during source
7 data processing, as represented by diamond 2596. If no errors were found, the normalization
8 process terminates (box 2580). If any errors are found, all of the items and attributes
9 determined as needing correction are scheduled for manual cleansing (or manual correction),
10 represented by box 2585, and the automatic cleansing terminates (box 2580).

11
12 The exact mechanism used to schedule manual cleansing and pass control to it while
13 concurrently continuing processing of the parts of the dataset that are not in error is not
14 important. There are many techniques known to the art which can be used for these purposes.

15
16 Fig. 17 shows the process of correcting validation errors, a manual validation process which
17 is represented by box 2129 in Fig. 12A.

18
19 Box 2605 represents commencement of manual validation. The first thing done, represented
20 by box 2615, is receipt of the list of validation errors. When these errors are received, the
21 activation of the manual validation process is recorded in the ETSDT. After this an error
22 entry is extracted, as represented by box 2620. Decision diamond 2625 represents the
23 identification of the error entry as either a source item or an attribute. If this error entry is for
24 a source item all of the associated attributes and any other relevant information are collected,
25 as represented by box 2630. Otherwise all the attributes that have the same source item and
26 are in question and any other relevant information are collected, as represented by box 2665.
27 The collection represented by box 2655 is a set of attributes with errors all of which are
28 associated with the same item, but the item is not included as it does not contain any errors.
29 As represented by box 2630, if the item has errors all of its attributes, with or without errors,
30 are collected. This is done since, in some instances, the item error affects the attribute
31 processing. In either case human assistance is requested, represented by box 2635, and the
32 identity of the human working on the errors is recorded in the ETSDT. The information is
33 passed to that person who corrects the errors. The manual correction process waits until the

1 error is, box 2640 and then records the corrections in the ETSDT. The process to continues
2 and checks to see if there are additional errors, a query represented by decision diamond
3 2645. If there are additional errors, the next error entry is extracted. Otherwise, all the errors
4 have been corrected, which means validated, so processing proceeds and the validated items
5 and attributes are scheduled for automatic normalization, as represented by box 2650. Lastly,
6 manual validation terminates (box 2655).

7
8 Fig. 18A shows the process of correcting normalization errors, a manual normalization
9 process which is represented by box 2114 in Fig. 12A. Box 2705 represents commencement
10 of manual normalization, with receipt of the list of normalization errors. The activation of the
11 manual normalization process is recorded in the ETSDT. After this an error entry is extracted,
12 as represented by box 2715. Decision diamond 720 represents the identification of the error
13 entry as either a source item or an attribute. If this error entry is for an item all of the
14 associated attributes and any other relevant information are collected, as represented by box
15 2725. Otherwise all the attributes that have the same item and are in question and any other
16 relevant information are collected, as represented by box 2727. The collection represented by
17 box 2727 is a set of attributes with errors all of which are associated with the same item, but
18 the item is not included as it does not contain any errors. As represented by box 2725, if the
19 item has errors all of its attributes, with or without errors are collected. This is done since, in
20 some instances, the item error affects the attribute processing. In either case human assistance
21 is requested, represented by box 2730, and the identity of the human working on the errors is
22 recorded in the ETSDT. The information is passed to the person who corrects the errors. The
23 manual correction process waits until the error is corrected, box 2735, and then records the
24 corrections in the ETSDT. The process to continue and checks for additional errors, a query
25 represented by decision diamond 2740. If there are additional errors, the next error entry is
26 extracted. Otherwise, all the errors have been corrected, which means normalized, so
27 processing proceeds and the normalized items and attributes are scheduled for automatic
28 cleansing, as represented by box 2745. Lastly, manual normalization terminates (box 2750).

29
30 Fig. 18B shows the process of correction cleansing errors, a manual cleansing process which
31 is represented by ellipse 2123 in Fig. 12A. Box 2760 represents commencement of manual
32 cleansing, with receipt of the list of cleansing errors. The activation of the manual cleansing
33 process is recorded in the ETSDT. After this an error entry is extracted, as represented by box

1 2765. Decision diamond 2770 represents the identification of the error entry as either a
2 source item or an attribute. If this error entry is for an item all of the associated attributes and
3 any other relevant information are collected, as represented by box 2775. Otherwise all the
4 attributes that have the same item and are in question and any other relevant information are
5 collected, as represented by box 2772. The collection represented by box 2772 is a set of
6 attributes with errors all of which are associated with the same item, but the item is not
7 included as it does not contain any errors. As represented by box 2775, the item has errors,
8 and all of its attributes, with or without errors are collected. This is done since, in some
9 instances, the item error affects the attribute processing. In either case human assistance is
10 requested, represented by box 2780, and the identity of the human working on the errors is
11 recorded in the ETSDT. The information is passed to the person who corrects the errors. The
12 manual correction process waits until the error is corrected, box 2785 and then records the
13 corrections in the ETSDT. The process to continue and checks for additional errors, a query
14 represented by decision diamond 2790. If there are additional errors, the next error entry is
15 extracted. Otherwise, all the errors have been corrected, which means cleansed, so manual
16 cleansing terminates (box 2795).

17
18 Fig. 19 shows a flowchart of the generic framework used to implement a cross-source process
19 which is represented by box 2138 in Fig. 12B. Recommended value is an example of a cross-
20 source process. This description illustrates application of a cross-source process after single-
21 source cleansing is complete. This is the advantageous embodiment. However, it is possible
22 to apply cross-source processes at different stages if required.

23
24 Ellipse 2800 represents commencement of processing commences when all of the candidate
25 datasets are ready for processing. Standard techniques initiate a cross-source process when
26 the source datasets are ready. First, all of the cleansed candidate source datasets are opened,
27 as represented by box 2802. Next, box 2804 represents the recording of all referenced
28 datasets. If the output is a new dataset, this will require the creation of ETSDTs for the new
29 dataset. If the output is an update to an existing dataset produced by the same process then the
30 existing dataset ETSDTs of are updated. All of the rules for the cross-source process are
31 acquired, as represented by box 2806. Box 2808 is the beginning of a loop where on each
32 iteration an item is extracted from all datasets that contain it.. If a new dataset is created, a
33 new ETSDT is created for this new item, and the dataset containing the item is recorded in

1 the ETSDT, as represented by box 2810. Box 2822 represents application of a rule to the
2 available items, which produces a new item value. The purpose of cross-source processing is
3 to produce values. Sometime new values are produced which did not previously exist. Other
4 processes produce their values by selecting one of the previously known values. Cross-source
5 processing result in new values by either method. If the item passes rule application,
6 represented by diamond 2820, then additional rules are checked (diamond 2823). If more
7 rules are found, the rules are applied (box 2822).

8
9 If the new item does not pass the rule application, the error and the attempt to correct it are
10 recorded, as represented by box 2830. Next, diamond 2815 represents performance of a check
11 to see whether the correction was successful. If the correction is successful, the new value
12 and the rule used for the correction are recorded in the ETSDT, as represented by box 28216.
13 If the correction was not successful, then the current value is flagged for intervention, as
14 represented by box 2835. In either case, successful or non successful correction, processing
15 proceeds to a check for more rules, a query represented by diamond 2823.

16
17 In cases where attribute level processing is involved, when no additional rules are found, box
18 2824 represents extraction of an attribute from all datasets that contained the extracted item.
19 The attribute and all datasets that contained it are recorded in the ETSDT, as represented by
20 box 2828. If this attribute is being created for a new dataset then a new attribute ETSDT is
21 created at this point. If this attribute is updated in an existing dataset, then the recording is
22 done to the ETSDT of the existing dataset. Sometimes for an existing dataset a new attribute
23 is found which results in the creation of a new ETSDT. Next, a rule is applied, represented by
24 box 2826. Success or failure of the rule application is represented by diamond 2840. If the
25 attribute passes the rule application, processing checks for additional applicable rules,
26 represented by diamond 2845. If additional rules are found, the next rule is applied box 2826.
27 If the attribute did not pass the rule application, represented by diamond 2840, the error is
28 recorded (box 2875) and a correction is attempted. Success or failure of the attempted
29 correction is represented by diamond 2876. If the correction is successful, then all of the rules
30 use to correct the attribute and the new attribute value are recorded in the ETSDT, as
31 represented by box 2877. If the correction was not successful, then the attribute is flagged for
32 intervention, as represented by box 2878. In both cases, successful or non successful,
33 correction processing proceeds to check for more rules (box 2845).

1 If no additional rules are found, processing checks for additional attributes, as represented by
2 decision diamond 2850. It is worth noting that it is not assumed that all source datasets have
3 the same attributes associated with each item when they contain the same item. More
4 attributes will continue to be processed until all of the attributes in each of the source datasets
5 have been processed. However, each attribute is processed once no matter how many source
6 datasets it occurs in.

7
8 If no additional attributes are found, processing checks for more items, as represented by
9 diamond 2855. It is worth noting that it is not assumed that all source datasets contain the
10 same items. The result of the query represented by diamond 2855 is true as long as any items
11 remain in any source dataset. However, each item is processed once, no matter how many
12 source datasets contain it. Effectively, each item is marked as processed in every source
13 dataset that contains it once it is found in one of them. Once all items have been exhausted,
14 by the query represented by diamond 2855, processing proceeds to check for errors,
15 represented by diamond 2860. If any items or attributes have been flagged as needing
16 intervention, manual cross-source correction is scheduled, as represented by box 2865. This
17 process is similar to single-source correction in that it request human intervention to correct
18 the error. The scheduling of the process, the human who intervenes and the values produced
19 are all recorded in the ETSDT. After manual cross-source correction has been scheduled, the
20 cross-source process terminates (box 2870). If no errors were found the cross source process
21 terminates (box 2870).

22
23 This concludes the description of the flow diagrams for this data cleansing and quality
24 enhancement aspect of the invention. In our preferred embodiment workflows are used to
25 implement the process and flows described herein. Alternative embodiments use script,
26 discrete distributed process, or a mixture of all of these. Any suitable mechanism or
27 programming language is used to implement the flows and processes described herein.

28 29 D. On-demand Dataset Delivery Processing

30
31 This aspect of the invention provides a flexible scalable multi-tenant information retrieval
32 and delivery system that supports multiple independent client organizations each having their
33 own data interests, data entitlements and data delivery requirements. This aspect of the

1 invention effectively enables a data delivery mechanism that interacts with a single repository
2 to serve multiple clients and/or requesters, even though each requester is only entitled to
3 some subset of the data in the multi-source multi-tenant data repository (further referred to as
4 “repository”) or, in a broader context, of the reference data available from the reference data
5 utility.

6
7 Requests for information retrieval and delivery are presented by requestors as a request for
8 the production and delivery of an on demand dataset. The specification of an on demand
9 dataset allows the requester to control (1) the information to be supplied in the dataset, (2)
10 preferences on which information sources to use in supplying values for the selected
11 information elements, (3) the mode of the data delivery, (4) the format of the data when
12 provided and (5) communication and data transfer control information for establishing
13 connections with the requester and effecting delivery. The data to satisfy an on demand
14 dataset request is retrieved by the method described above in section B for multi-source
15 multi-tenant data repository. Enforcement of data entitlements – ensuring that requestors
16 never receive values from information sources to which they are not entitled – is provided
17 either by the repository or by additional logic in the on demand dataset delivery processing.
18 Delivery modes supported by the invention include (1) on demand datasets which may
19 consist of a single one time delivery instance as needed for an ad-hoc query, (2) recurring
20 batched delivery instances and (3) quasi real time delivery.

21
22 The described apparatus and method for on demand dataset delivery supports multiple
23 customers with each customer having multiple requests for on demand datasets concurrently
24 outstanding. The method is flexible and able to support a wide range of requester delivery
25 and retrieval requirements because different aspects of this task have been separated out into
26 separate specification units of the on demand dataset request specification. The method is
27 scalable to allow concurrent processing of multiple requests and to support multiple
28 requesters with multiple requests from each because it exploits this separation of concerns to
29 allow automated processing on demand dataset requests. Each arriving on demand dataset
30 request has its specification automatically compiled into an on demand dataset production
31 process which is then executed to retrieve the required data and deliver it to the requester.
32 The invention supports any combination of allowed specifications for each of the separate on
33 demand dataset aspects listed above.

1 This aspect of the invention also provides the capability for the customer to specify the output
2 format for delivery of the data in customer specific format or an industry standard format.

3 The invention allows for delivery of information to a customer to take the form of loading the
4 identified data into a data mart own by that customer. This invention provides audit and
5 logging capability to ensure complete process transparency, non-repudiation, billing and
6 other auditing purposes.

7
8 The method is effectively an on demand approach to data delivery for reference data. The
9 ability to support a wide range of client requirements for different topics, sources, qualities,
10 modes and formats, organized as an automated extensible system provides a valuable service
11 by enabling the complex but critical delivery functions to be centralized and highly
12 leveraged.

13
14 The described invention supports customer and data source privacy. Since independent
15 production processes are generated for each on demand dataset request, and data entitlements
16 are enforced, no customer or data source is able to discover information about another's data,
17 queries or other actions to retrieve and deliver information to them.

18
19 The method is described herein as it applies to reference data used by Financial Services
20 businesses. This method for enabling flexible scalable delivery of on demand datasets in the
21 context of a multi-source multi-tenant data repository 20, as described above, has many other
22 possible areas of application. The multi-source multi-tenant data repository 20 manages and
23 provides permanent storage for repository information elements, associated metadata,
24 entitlements, value add functions and documents. Access to consumer credit information,
25 government regulation and registration information, and telecommunications usage
26 information are three additional examples where the method has use. Characteristics of
27 contexts where the method has use and of reference data are: (1) the information comes from
28 many sources; (2) there are multiple users, potentially in independent organizations, that need
29 access to the same information but potentially with different source entitlement rights; (3) the
30 referenced information is accessed by users largely in read-only mode except when they
31 participate in correcting invalid values; (4) high quality timely information is both valuable
32 and complex to gather, hence the efficiencies from a utility approach, shared infrastructure
33 and shared data quality enhancement provide significant benefit; and (5) entitlement

1 enforcement and privacy management must be provided by such a utility. Although the
2 invention is described in the context of financial services reference data, which is one
3 important area of application, the approach revealed herein, enabling an effective utility to
4 provide data access meeting the requirements above, has value in any context with these
5 requirements.

6
7 Fig. 20A is a flow chart for producing an on demand dataset in response to an on demand
8 dataset request. Box 3100 in this figure is the outer box representing the overall method. In
9 the context of a reference data utility this corresponds to the client data delivery processing
10 first introduced as block 21 in Fig.1A. The initial step in this flow chart, box 3101,
11 represents receipt of a single on-demand dataset request to produce a single on demand
12 dataset.

13
14 Box 3101 represents receipt of the on demand dataset request. This invention does not
15 specify the type of channel through which the request is passed. The invention defines the
16 content of the requests and allows the input request to be formatted in a manner that is
17 consistent with the way it is delivered. The invention supports the receipt of requests via any
18 number of communication protocols and semantics. Requester authentication and
19 authorization is handled in this step with unauthorized requests logged and discarded. Valid
20 requests are saved in an internal form as represented by data element 3116, which is
21 described in more detail in Fig. 22A. Receipt of on demand dataset requests is also logged for
22 traceability and non-repudiation purposes.

23
24 The dashed line connecting box 3101 with data element 3116 shows that the on demand
25 dataset request specification is received as part of the on demand dataset request received in
26 box 3101. The on demand dataset request specification represented by data element 3116 is
27 available as input during subsequent processing steps.

28
29 Box 3102 represents the actions of parsing, validation and analysis of the on demand dataset
30 request specification (data element 3116) received in the on demand dataset request. The
31 parsing, validation and analysis step is described in more detail in Fig. 20B. This is followed
32 by box 3103, which represents the action of setting up the process to produce the on demand
33 dataset. This process is created by assembling a workflow process out of parameterized

1 activity building blocks. An alternative embodiment is to accomplish this by parameterizing
2 the parts of a workflow used for all on demand datasets. Anyone skilled in the art
3 understands the technologies needed to build a script or workflow for a pre-specified task,
4 either statically or dynamically. The processing represented by box 3103 is described in more
5 detail in Fig. 21A. Box 3104 represents the execution of the on demand dataset production
6 process assembled and deployed, as represented by box 3103; this will produce the requested
7 dataset and deliver it to the requester. Decision box 3105 shows that the outer structure of the
8 method is a loop; after processing an on demand dataset request, control loops back and
9 logically handles the next request for an on demand dataset.

10
11 Fig. 20A shows the simplest logical form of the method in which requests for on demand
12 datasets are handled sequentially in a single loop. An advantageous embodiment extends this
13 representation using concurrency techniques well understood to those skilled in the art to
14 allow multiple instances of the loop formed by boxes 3101, 3102, 3103, 3104, and 3105 to be
15 handled concurrently. Such an extension enables the method to handle multiple requests for
16 on demand datasets simultaneously.

17
18 The on demand dataset requests are able to modify or terminate the results of previous on
19 demand dataset requests. This is handled as a dynamic replacement or termination of the
20 process created as a result of the previous request. How to schedule these requests, or where
21 to schedule them or building schedulers which allow termination or replacement of
22 previously scheduled tasks is not the focus of this invention. These functions are well known
23 to those skilled in the art.

24
25 Fig. 20B shows a flowchart of the steps in the parsing and analysis of an on demand dataset
26 request specification, describing in more detail the action represented by box 3102 from Fig.
27 20A where an on demand dataset request specification is parsed, analyzed and validated.

28
29 The outer box of Fig. 20B is box 3102 which was first introduced in Fig. 20A. The output of
30 the parse and analyze step is a parsed block of data representing the information in the
31 specification but now organized for assembly of a process tailored to produce exactly the
32 requested data. Box 3106 represents the initialization step to set up an empty output structure
33 into which parsed blocks can be added. The on demand dataset request specification is a

parameter block or text structure which is organized as a number of lexically distinct sections or stanzas, each dealing with a specific aspect of the on demand dataset. Each stanza is expected to contain information about an aspect of the on demand dataset. Box 3107 obtains the next stanza of the input specification and is the heading block of the stanza processing loop. Decision box 3108 resolves the stanza type. The key stanza types are: select data process, the sourcing policy, the delivery mode specification, data output format choices, and data delivery and transport characteristics. The stanza types and the information provided in each stanza type are discussed in more detail in Figs. 22A and 22B. Boxes 3109, 3110, 3111, 3112, and 3113 provide different parsing analysis and validation logic for each of these stanza types. Although these stanzas represent the key required aspects of an on demand dataset request specification, additional stanza types are possible. The architecture of this component is extensible. In an alternative embodiment requestor specific stanza types are allowed. The result of the stanza type specific parsing is a parsed output block. Box 3114 in the flow shows that on completion of the stanza type specific parsing, the resulting parsed output block is added into the output. Decision box 3115 tests whether the on demand dataset request specification has been completely processed or whether there are additional stanzas still to be parsed. If more stanzas are available to be parsed, control loops back to box 3107 to process the next one. If the input specification is fully parsed, control flows out of box 3102 and parsing, analysis and validation are complete.

An important aspect of the on demand dataset processing is that each distinct aspect of the on demand dataset is specified and then parsed separately. This separation of concerns enables on demand datasets to meet a wide range of data selection and delivery needs required to provide delivery of data to many customers from within a shared multi-source multi-tenant data repository. An advantageous embodiment of the method described herein provides initial elaborations of options for each of these aspects. Simple extensions of the method are made by providing richer options in each of these independent aspects of an on-demand dataset.

Data element 3116, originally introduced in Fig. 20A, is a representation of the data structure used by the requester to supply the on demand dataset request specification. This specification is the input to the parsing, analysis and validation processing represented by box

1 3102. The data structure of the on demand dataset request specification is elaborated in Figs.
2 22A and 22B.

3
4 Data element 3117 represents the parsed on demand dataset specification produced as output
5 from the flow of box 3102. This parsed specification is used as input in Fig. 21A where the
6 customized on demand dataset workflow for producing the specified on demand data set is
7 assembled.

8
9 Fig. 21A is a flowchart that shows the steps in setup of a customized on demand dataset
10 production process, describing in more detail the action represented by box 3103 that was
11 introduced in Fig. 20A. This is the step of assembling and deploying a customized on demand
12 dataset production process tailored to the requirements of a parsed on demand dataset request
13 specification, as represented by data element 3117.

14
15 The flow starts with box 3201 in Fig. 21A, in which the next available block from data
16 element 3117 is picked up. Box 3202 locates the matching activity building block from a
17 library of available activity building blocks. The library is represented as data element 3210
18 and is described in more detail in Fig. 21B. Box 3203 represents the action of applying the
19 information and parameters obtained from data element 3117 to the matching activity
20 building block to produce a specific activity tailored to provide the exact function needed for
21 this phase of the process to create the requested on demand dataset. Box 3204 saves this
22 tailored activity so that it is available subsequently for assembly into a complete process.
23 Decision box 3205 is a test to determine whether all blocks in the parsed data have been
24 handled and had tailored activities produced for them. If not, control loops back and resumes
25 at box 3201 for the next iteration.

26
27 Box 3206 is reached when all parsed specification information has been processed and
28 converted into a set of parameterized (tailored) activity blocks. The processing represented
29 by box 3206 is to sort these activity blocks into the correct order, insert default activity
30 blocks for any phases for which no specification has been supplied and provide an overall
31 flow of control yielding a set of tailored activities which is the basis of the on demand dataset
32 production process. Box 3207 involves adding specific listeners into this process.

1 Listeners are needed if the process has to be sensitive to the arrival of new information in the
2 multi-source multi-tenant data repository from which data elements are being selected for the
3 on demand dataset. The presence of listeners makes the on demand dataset production
4 process sensitive to execution time control commands from the user such as prompts for
5 when additional data is to be delivered. An alternate embodiment is for the attachment of
6 listeners to be included in individual building blocks from the library of activity building
7 blocks and to parameterize these listener functions for the specific connection needed. Any
8 technique for enablement of asynchronous receipt of information is applied to enable these
9 listeners.

10
11 Although the stanzas and library of building blocks described herein represent the key
12 required aspects of an on demand dataset request specification, additional stanza types are
13 also possible.

14
15 Box 3208 represents the action of deploying the assembled on demand dataset production
16 process so that it is ready to be executed for run time production and delivery of the requested
17 on demand dataset. This is shown with a dashed arrow to box 3104. Box 3104 is described in
18 more detail in Figures 23A and 23B

19
20 After completion of the activities represented by box 3208, control flows out of box 3103.
21 Initiation of the deployed process is represented by box 3104 of the top level flow in box
22 3100 described in Fig. 20A.

23
24 Techniques such as workflow processing, well known to those skilled in the art, are used to
25 implement and manage the generated on demand dataset production process.

26 An advantageous embodiment of this process represented by box 3103 tailors the same basic
27 process template to produce a specified process, customized to produce the requested on
28 demand dataset. An alternative embodiment, obvious to those skilled in the art, is to generate
29 a separate process for each on demand dataset request using the same phase by phase
30 construction process. Another alternative is to use parameterized static workflows. Another
31 embodiment is to use a compiler. Those skilled in the art realize that there are many
32 technologies that can be used to produce the process which produces the on demand dataset.
33 The appropriate scheduling mechanism is used in box 3104.

Fig. 21B shows the contents of the library of activity building blocks. The library of basic activity building blocks was introduced as data element 3210 in Fig. 21A. Basic activity building blocks are provided for each of the main phases of the on demand dataset production process. Box 3212 shows the activity building block for the item selection phase; box 3213 shows the activity building blocks for the sourcing policy; box 3214 shows the activity building block for the delivery mode; box 3215 shows the activity building block for the delivery and transport phase and box 3216 shows the activity building block for the output format phase.

The specific capabilities of each of these activity building blocks are described in more detail in Figures 23A and 23B wherein the steps and phases of the on demand dataset production process that produces and delivers an on demand dataset are elaborated.

In an alternative embodiment, additional activity building blocks are added into the library. An example of an additional activity building block is a special activity building block to handle the loading of a customer datamart with the information in the on demand dataset instead of just delivering the data to the requester as described herein. In another embodiment these processes are factored in a way to distribute part of this processing to the requester or increasing the number of activity building blocks or decreasing the number of activity building blocks. The point of this invention is that these processes occur; the exact factorization used in any specific implementation is left to those skilled in the art.

Fig. 22A shows the organization of an on demand dataset request specification. The request represents a single request specification from one requester. The method allows a single person, application or organization making requests to have multiple on demand dataset requests outstanding concurrently. From the perspective of the delivery method there is no difference in the processing of multiple concurrent on demand dataset requests from a single end user and multiple concurrent on demand dataset requests from independent end users.

The separate components of an on demand dataset request specification are shown as boxes 3301 – 3305, each of which is described in detail below. Each of these sections of an on demand dataset specification is a separate stanza which can be parsed and processed by a separate iteration of the parse processing as represented by box 3102 in Figure 20B. The

1 components of the on demand dataset request specification described herein represent the key
2 required aspects necessary for the successful assembly and delivery of the on demand dataset.
3 Additional aspects specified in the specification are also possible.
4

5 Box 3301 represents the select data specification unit. This specifies the information elements
6 whose values are to be delivered in the requested on demand dataset. The specification unit
7 is in the form of a filter or query against the repository entity metadata and properties using
8 predicates on topic, subtopic and other attributes and values of the repository entity.

9 Specifically, the filter determines the repository entities of interest and the properties and
10 attributes of those repository entities for which values are to be returned in the dataset. The
11 selection criteria include any reasonable criteria by which items are selected, such as interest
12 lists, temporal constraints, various classifications, etc. A relational query is one possible
13 implementation. The requester receives one or more current values from the set of entitled
14 available current values for each selected attribute or property of each selected repository
15 entity.
16

17 Box 3302 represents the source policy specification unit, sometimes called source preference,
18 where a source preference can be specified. The preferred embodiment uses a simple
19 preference order on sources and item instance processes producing attribute values. If there
20 is a choice of available values entitled to this requester for a specific element, the first such
21 value in the supplied preference order is used. In addition to actual data origins, item instance
22 processes appear in this preference order. For example, the requester specifies a preference
23 order between explicitly using a particular data origin and using a recommended value
24 derived by some input cleansing and enhancement process that selects a value after
25 comparing the values received from multiple data origins. In an alternative embodiment, a
26 default ordering on sources is provided to handle the case where this was not specified by the
27 requester.
28

29 Another alternative embodiment supplies a more sophisticated sourcing policy that is
30 sensitive to the information element on which it applies. This policy specifies a conditional
31 source preference ordering, subject to a predicate on the properties, attribute values or
32 metadata of the information element. For example, in a financial reference information
33 context, a requester specifies that source A is preferred to source B on common stocks but

1 that source B is preferred to source A on public and government bonds. Preferences are
2 flexibly described through the predicates. A requester expresses a preference, for example,
3 for particular sources for stocks traded on a specific exchange, or that recently arriving or
4 unconfirmed data from a particular source could be discounted.

5
6 An alternative embodiment of sophisticated sourcing policy uses a set of rules, each with the
7 form of a simple preference order or a conditional preference sensitive to values in, and
8 properties of, the item as described above. When applying the sourcing policy to select
9 values for inclusion in the on demand dataset, these rules are evaluated in turn by the
10 sourcing policy step and the resulting preferred value selected.

11
12 Box 3303 represents the delivery mode specification unit. The delivery mode is a feature that
13 gives on demand datasets significant flexibility to respond to different requester
14 requirements. It allows the requester to create on demand datasets with a single one-time
15 delivery instance or on demand datasets with recurring delivery instances. A more complete
16 description of the delivery mode is provided in Fig. 22B below.

17
18 Box 3304 represents the delivery and transport specification unit. The customer supplies
19 information governing connection and communications protocols and the authentication
20 checks required for each delivery instance in the on demand dataset. The dataset delivery and
21 transport specification unit also provides network addressing, protocol and authentication
22 information needed to establish a connection for each delivery instance. This includes
23 "outbound" connection and authorization specifics used to initiate delivery instance
24 connections from the repository and delivery method to the requester. It also includes
25 inbound connection and authentication information to allow the requester to connect in and
26 initiate a delivery instance. If an outbound connection is specified, the requester defines
27 where and how the connection is to be set up; if the connection is inbound, it specifies the
28 necessary authentication. In either case the file or data transfer protocol used to pass the
29 delivery dataset is specified. A datamart is specified as the target of delivery with the
30 requester supplying appropriate database load parameters. Technologies such as table
31 replication mechanisms are then applicable in enabling this transport option.

1 In an advantageous embodiment described herein, the scheduling information governing
2 exactly when the next delivery instance of an on demand dataset occurs is provided in the
3 specifics of the delivery mode specification unit. An alternative embodiment packages this
4 information with the dataset delivery transport specification unit.

5
6 Box 3305 represents the output format specification unit, which allows the requester to
7 specify data formats and transformation rules governing the delivery format of the on demand
8 dataset and its contained information elements. Each information element in the repository
9 has one or more preferred data output formats. For example, when adding financial
10 instrument data to an on demand dataset, a public standard such as Market Data Description
11 Language (MDDL) or the ISO financial instruments structure 20022 is used. The output
12 format unit allows the requester to choose between standard formats or to specify some
13 customized format.

14
15 Part of the value of on demand dataset request specification is that the specification is
16 structured as separate units, allowing for separation of concerns.

17
18 Fig. 22B shows the on demand mode case tree, elaborating the different delivery modes
19 introduced in Fig. 22A. As such, it is an expanded description of box 3303, which represents
20 the delivery mode specification unit. Fig. 22B is a tree structure with lower levels of the tree
21 being sub-cases of their parent element. Box 3306 is the root node representing delivery
22 modes. An on demand dataset has either a one time delivery, as represented by box 3307, or a
23 recurring delivery, as represented by box 3308.

24
25 Box 3307 represents one time delivery. An on demand data set with one time delivery mode
26 is produced by applying one or more retrieval operations to the current state of the repository,
27 assembling the retrieved information in and delivering it to the requester as the single
28 delivery instance for this on demand dataset.

29
30 Box 3308 represents recurring delivery. An on demand dataset with recurring delivery mode
31 specifies that multiple delivery instances are requested. Each delivery instance represents a
32 separate retrieval of information from the repository. The exact method used to accumulate
33 the data is determined by other predicates. The delivery dataset returned to the requester in

1 each delivery instance contains information that has been retrieved over time and
2 accumulated in a delivery dataset in preparation for use with the next delivery instance of this
3 on demand dataset. Alternatively, a delivery data set is created when it is needed for delivery
4 by applying one or more retrieval operations on the state of the repository at that time.

5
6 A recurring delivery is either a batched delivery, as represented by box 309, or a quasi-real
7 time delivery, as represented by box 3310. Box 3309 represents batched delivery. Processing
8 for each delivery instance is done by making the delivery method aware of new information
9 arriving in the repository, by periodic retrieval operations on the repository or by a retrieval
10 action on the state of repository at the time the delivery dataset is needed. Box 3310
11 represents quasi-real time delivery mode. This is a case of recurring delivery mode where
12 relevant new arriving information is delivered to the requester as soon as it is detected. This
13 typically leads to a fine grained sequence of delivery instances with each delivery dataset
14 containing only a small amount of data. The term quasi-real time is used since providing
15 updated information in frequently updated transfers is the key characteristic.

16
17 This completes the description of the main delivery modes. Boxes 3311, 3312, 3313, 3314
18 and 3315 represent additional parameters that can be applied to boxes 3309, 3310 and 3307.
19 For simplification purposes they are described in the context of box 3309.

20
21 Box 3311 represents a prescheduled batch where there is a fixed predetermined schedule
22 controlling when the delivery instance occurs. Box 3312 represents the case of on demand
23 delivery instances. In this case the requester explicitly requests that the delivery instance be
24 instantiated and delivered. The requester also indicates when the next delivery instance is
25 required. Box 3313 represents the case of data driven delivery which is based on some
26 function of the state of the data, such as the volume of data, or arrival of particular data
27 elements.

28
29 A delivery instance contains either a complete set of all selected values or only new and
30 changed values since the last delivery instance (or over some period of time). These two
31 options are represented by boxes 3314 and 3315, respectively. These options are represented
32 as sub-cases of prescheduled batched delivery mode, represented by box 3311, but they can

1 obviously be applied to boxes 3312 and 3313. The usefulness varies depending upon the
2 context.

3
4 Alternative embodiments include an on demand mode that allows the requester to specify that
5 the selected information elements be loaded into a private working database or datamart set
6 up exclusively for that requester's use. The choice of a datamart for delivery influences the
7 delivery transport specification. In a one-time query, the on demand mode indicates whether
8 additional research and data gathering is to be launched to gather new values in the event that
9 there is no appropriate value currently in the repository for a specified information element.
10 Additional modes include an alert mode, in which event notices are sent if the value of some
11 reference item crosses a pre-specified threshold, or a summary report mode, in which
12 aggregated summary reports on reference item values sets are sent at specified intervals.

13
14 Fig. 23A describes the flow of an on demand dataset production process used at runtime to
15 produce an on demand dataset and deliver it to the requester. This process was first
16 introduced in Fig. 20A, represented by box 3104. Fig. 21A explains how a customized on
17 demand dataset production process is generated to meet the requirements of a particular on
18 demand dataset specification. As previously noted, the effect of executing an on demand
19 dataset production process is to retrieve information from a repository subject to the
20 requester's selection and sourcing specification, assemble this information into a delivery
21 dataset subject to the requesters, delivery mode and format specification, then delivering the
22 data to the requester subject to their dataset delivery and transport specification.

23
24 Control enters box 3104 in Fig. 23A from the top and first passes to box 3401 where
25 processing of the next delivery instance is started. This reflects the fact that recurring on
26 demand datasets are delivered to the requester as sequence delivery instances. The outer
27 control structure of the flow to produce an on demand dataset is a loop; each iteration of this
28 loop results in the production of one delivery dataset transferred to the requester as one
29 delivery instance.

30
31 The next step in the flow is represented by box 3402, where processing of the next
32 information element is started. The inner control structure of the flow to produce the next

1 delivery instance of an on demand dataset is a loop; each iteration of the loop will add one
2 information element into the delivery dataset.

3
4 The next step in the flow is represented by box 3403. This step retrieves and formats one
5 information element from a multi-source multi-tenant data repository. Elements are only
6 retrieved if the requester is entitled to the information. The retrieved element is inserted into
7 an accumulating delivery dataset. As noted by the dashed line connecting this box to data box
8 3407, this step uses information from the repository. That repository could be an entitlement
9 enforcing repository as described in section B or more broadly in the context of a reference
10 data utility the entitlement managed entity data, box 50 in Fig.1A. More detail on the
11 processing of box 3403 is provided in Fig. 23B below.

12
13 The next step in the flow is represented by decision box 3404 which results in the flow either
14 terminating the element loop and moving on to delivery instance processing or returning to
15 box 3402 to add the next information element into this delivery dataset. When there are no
16 more elements, control passes to box 3405, execute delivery instance. This is the processing
17 to take all information elements which have accumulated in the temporary delivery dataset
18 waiting for a delivery instance, organize them into a delivery instance and transfer them to
19 the requester. The logic for this is described in greater detail in Fig. 23C below.

20
21 Finally, box 3423 represents a query for additional delivery instances and, if one is found,
22 schedules the next delivery instance in the case of continued datasets. Box 3401 is scheduled
23 with a pointer (or reference) to the parsed on demand dataset request specification. Whether
24 or not anything is scheduled is determined by the delivery mode of the on demand dataset. If
25 the on demand dataset is on-time and has been completely delivered by preceding data
26 delivery instances, nothing is scheduled. If more instances are needed to complete the
27 delivery of currently available data, or, the on demand dataset is recurring and the delivery
28 mode is not on demand, box 3401 is scheduled immediately. If the on demand dataset is
29 recurring and the delivery mode is on-demand then a listener is also activated to wait for the
30 next delivery request. When the listener receives the request it schedules the immediate
31 execution of box 3401.

1 As noted elsewhere, a user request is used to terminate an existing recurring on demand
2 dataset. When such a request arrives, either the next scheduled instance is terminated or,
3 because it is active, a flag is set indicating that no more requests are to be allowed. Finally,
4 control flows out of box 3104; execution of the workflow producing the on demand dataset is
5 complete.

6
7 Fig. 23B shows a flowchart that elaborates the processing represented by box 3403
8 introduced in Fig. 23A, retrieving a new information element and adding it into the delivery
9 dataset of accumulated values waiting for delivery to the requester.

10
11 The first step in this flow is represented by box 3410, which locates the repository entity
12 containing the new information element. In general, the element selection unit of the dataset
13 specification (box 3301 in Fig. 22A) provides property values such as entity name or entity
14 topic which enables the relevant entity to be located in the repository. Parsing and process
15 assembly of the dataset request specification in boxes 3102 and 3103 of Fig. 20A have
16 converted its item select unit into a specific selection operation on the repository, which
17 returns the entity.

18
19 In addition to selecting a specific repository entity, the element selection unit of the dataset
20 specification indicates which attributes or properties of that entity are returned in the dataset.
21 Requesting all available attributes or all properties is a special case. The property and
22 attribute selection is compiled into repository operations, which are then executed in the
23 following step, represented by box 3411.

24
25 Box 3412, represents the step of gathering from the repository those values of the selected
26 properties and attributes of the selected entity that the requester is entitled to receive. This
27 processing requires knowledge of the entitlements of the requester and the sourcing of
28 information elements in the repository. It may involve gathering values from multiple item
29 instances of the selected repository entity. In an advantageous embodiment entitlement
30 enforcement is provided as a function of the repository. An alternate embodiment implements
31 an entitlement enforcement scheme as part of this processing block. As a result of the
32 processing of box 3412 the entitled set of values is gathered for the identified attributes and

1 properties of the selected entity. Any values that the requester specified to which the
2 requester is not entitled will not be included.

3
4 Box 3413 represents application of the sourcing preference rules specified in the source
5 preference unit (box 3302 in Fig. 22A). Hence, if multiple values with different sourcing are
6 available for a particular attribute the value from the source appearing earlier in the requester
7 preference list will be selected. Sourcing preference is specified as a preference between
8 identified item instances in the repository. For example, a requester can specify a preference
9 for values from a recommended value process over the values provided by a particular source
10 or vice versa.

11
12 An advantageous embodiment allows for multiple variations in the specification of sourcing
13 preferences. First, a source preference can be specified to apply only to a particular attribute
14 or property of particular entity. Or, a preference could be specified to apply uniformly over
15 all attributes of all selected entities in a dataset. Preference can also apply to one attribute of
16 all entities in a particular subclass. An example is the use of one preference on ratings of
17 municipal bonds but a different preference on all definition of common stocks. Finally, a
18 requester can specify that values from multiple entitled sources are included in the dataset
19 allowing the requester to make their own comparisons between the values from different
20 sources or repository processing. All of these functions are included in the processing of box
21 3403.

22
23 Control then flows to box 3414 where data format conversions are applied to the values
24 obtained from the repository following the format specifications from the requester provided
25 in box 3305 in Fig. 22A. This format processing is compiled into executable logic by
26 tailoring a formatting activity building block as part of the process assembly processing in
27 Fig. 21A. Requester specified transformation rules are applied to the on demand dataset to
28 convert it to the required delivery data format. For each category of provided data, the on
29 demand dataset delivery supports preferred data output formats for passing data values to the
30 requester. For example, when passing instruments data a public standard such as Market Data
31 Description Language (MDDL) or the ISO financial instruments structure ISO 20022 is used.

1 Finally, box 3415 adds the formatted selected values into the temporary dataset, which is
2 being accumulated for delivery to the requester in the next delivery instance. The on demand
3 mode of the dataset may also affect this processing step. If only new and changed values of a
4 pre-scheduled batched dataset are to be delivered, this step will only add the value to the
5 temporary dataset if this is a new or changed value since the last delivery instance.

6
7 After box 3415 processing is complete, control flows out of box 3403; a new information
8 element has been formatted and added into the accumulating data waiting for delivery to the
9 requester in the next delivery instance.

10
11 Fig. 23C shows a flow chart of the processing steps comprising execution of a delivery
12 instance originally introduced as box 3405 in Fig. 23A. This processing is responsible for
13 gathering the accumulated delivery dataset of selected, formatted values and transferring this
14 to the requester.

15
16 The outer box of Fig. 23C is box 3405; more detail on the processing of this block is
17 provided in the form of a flow chart. Control enters from the top and passes to the first step,
18 represented by box 3420, where final formatting of the accumulated delivery dataset is done
19 following format specifications provided in box 3305 of Fig. 22A. This formatting of the
20 complete accumulated dataset includes actions such as packaging up the entire dataset in a
21 particular way, adding summary and aggregated information. Formatting of the individual
22 information elements in the delivery dataset has been handled in an advantageous
23 embodiment of the step represented by box 3414 in Fig. 23B when the element was first
24 added into the accumulated data. Alternative embodiments relocate format processing
25 without changing the substance of this invention.

26
27 Box 3421 represents processing of the actual delivery and transfer protocols following the
28 specification provided in the step represented by box 3304 in Fig. 22A. This processing
29 involves establishing a network connection to the requester at some known network address,
30 authenticating on this connection and executing a file transfer protocol. Alternatively, it
31 involves returning data as a response parameter in a call setting up a one-time on demand
32 dataset request.

Box 3422 represents logging or creating an audit trail for this delivery. This capability ensures complete traceability of the on demand dataset. Non-repudiation services are provided to ensure the integrity of the on demand dataset. When use in the context of a reference data utility, client delivery logs as represented by box 29 in Fig. 1B would be updated as a result of this logging. After completion of this step, control flows out of box 3405. The delivery instance has now been executed.

This concludes the description of the flow and other diagrams for the on demand dataset delivery processing aspect of the invention. In a preferred embodiment workflows are used to implement the process and flows described herein. Alternative embodiments use script, discrete distributed process, or a mixture of all of these. Any suitable mechanism or programming language is used to implement the flows and processes described herein.

Published United States Patent Application 2005/0216416 of Abrams et al., entitled "Business Method for the Determination of the Best Known Value and Best Known Value Available for Security and Customer Information as Applied to Reference Data", and assigned to the same assignee as the present invention, is incorporated herein by reference in its entirety. This document is directed to a reference data facility that is structured to insure that no customer receives data or benefits from the knowledge of data content from a vendor with whom they do not have a contractual arrangement or to whose data they are otherwise not entitled.

The present invention can be realized in hardware, software, or a combination of hardware and software. It may be implemented as a method having steps to implement one or more functions of the invention, and/or it may be implemented as an apparatus having components and/or means to implement one or more steps of a method of the invention described above and/or known to those skilled in the art. A visualization tool according to the present invention can be realized in a centralized fashion in one computer system, or in a distributed fashion where different elements are spread across several interconnected computer systems. Any kind of computer system - or other apparatus adapted for carrying out the methods and/or functions described herein - is suitable. A typical combination of hardware and software could be a general purpose computer system with a computer program that, when being loaded and executed, controls the computer system such that it carries out the methods

1 described herein. The present invention can also be embedded in a computer program
2 product, which comprises all the features enabling the implementation of the methods
3 described herein, and which - when loaded in a computer system - is able to carry out these
4 methods. Methods of this invention may be implemented by an apparatus which provides the
5 functions carrying out the steps of the methods. Apparatus and/or systems of this invention
6 may be implemented by a method that includes steps to produce the functions of the
7 apparatus and/or systems.

8
9 Computer program means or computer program in the present context include any
10 expression, in any language, code or notation, of a set of instructions intended to cause a
11 system having an information processing capability to perform a particular function either
12 directly or after conversion to another language, code or notation, and/or after reproduction in
13 a different material form.

14
15 Thus the invention includes an article of manufacture which comprises a computer usable
16 medium having computer readable program code means embodied therein for causing one or
17 more functions described above. The computer readable program code means in the article
18 of manufacture comprises computer readable program code means for causing a computer to
19 effect the steps of a method of this invention. Similarly, the present invention may be
20 implemented as a computer program product comprising a computer usable medium having
21 computer readable program code means embodied therein for causing a function described
22 above. The computer readable program code means in the computer program product
23 comprising computer readable program code means for causing a computer to effect one or
24 more functions of this invention. Furthermore, the present invention may be implemented as
25 a program storage device readable by machine, tangibly embodying a program of instructions
26 executable by the machine to perform method steps for causing one or more functions of this
27 invention.

28
29 It is noted that the foregoing has outlined some of the more pertinent objects and
30 embodiments of the present invention. This invention may be used for many applications.
31 Thus, although the description is made for particular arrangements and methods, the intent
32 and concept of the invention is suitable and applicable to other arrangements and
33 applications. It will be clear to those skilled in the art that modifications to the disclosed

1 embodiments can be effected without departing from the spirit and scope of the invention.
2 The described embodiments ought to be construed to be merely illustrative of some of the
3 more prominent features and applications of the invention. Other beneficial results can be
4 realized by applying the disclosed invention in a different manner or modifying the invention
5 in ways known to those familiar with the art.

Claims

What is claimed is:

1. A reference data utility for serving a plurality of recipients, comprising:

data inputs for receiving unprocessed reference data from a plurality of sources;

a processor for processing the unprocessed reference data received so as to generate processed reference data having an increased value;

a repository for storing the unprocessed reference data and the processed reference data; and

an output generator for generating output data for delivery to recipients, in accordance with specifications of recipients; so that delivered output data contains at least one of unprocessed reference data and processed reference data, that said recipient is entitled to receive;

wherein the reference data utility is scalable so as to support an increasing number of sources and an increasing number of recipients

2. The reference data utility of claim 1, configured as a multi-tenant utility.

3. The reference data utility of claim 2, wherein the utility is implemented as a system of shared resources.

4. The reference data utility of claim 3, wherein said shared resources comprise at least one of the following: repositories, experts, processing, communications links, and data storage facilities.

5. The reference data utility of claim 1, further comprising means for tenants to perform self service administration of their clients.

6. The reference data utility of claim 1, wherein the repository further stores a plurality of business documents, and the output generator provides as output a selected group of said documents.

7. The reference data utility of claim 1, further comprising a memory portion for storing processed and unprocessed reference data and, with each unprocessed or processed reference

1 data element, a record of the data sources and applied processing used to derive said element;
2 said sourcing and processing determining the entitlement of individual recipients to receive
3 said element.

4
5 8. The reference data utility of claim 7, wherein the recipients are individuals granted
6 entitlement to particular sources of reference data and enhancement processes by at least one
7 of a plurality of tenant organizations sharing use of the reference data utility.

8
9 9. The reference data utility of claim 1, wherein the unprocessed reference data comprises
10 information elements, and the reference data utility further comprises means for annotating a
11 plurality of said information elements with sourcing information.

12
13 10. The reference data utility of claim 9, wherein the information elements have attributes,
14 and the reference data utility further comprises means for annotating said attributes with
15 sourcing information.

16
17 11. The reference data utility of claim 9, further comprising means for maintaining
18 information about entitlement of recipients to said information elements based on said
19 sourcing information.

20
21 12. The reference data utility of claim 1, comprised of components located in geographically
22 dispersed regions.

23
24 13 The reference data utility of claim 12, wherein components located in one of said
25 geographically dispersed regions are sufficient to operate as an independent reference data
26 utility.

27
28 14. The reference data utility of claim 13, wherein each independent reference data utility
29 includes a local repository, further comprising communication facilities for exchange of
30 information between said local repositories.

31
32 15. The reference data utility of claim 13, wherein each independent reference data utility is
33 specialized to provide information pertaining to a particular geographic region, and uses said

1 communication facilities to obtain and provide information from other independent reference
2 data utilities in other geographic regions.

3
4 16. The reference data utility of claim 1, further comprising an accuracy reporter for reporting
5 accuracy of processes performed by said reference data utility.

6
7 17. The reference data utility of claim 1, further comprising a configuration manager for
8 managing parameters of said reference data utility.

9
10 18. The reference data utility of claim 17, wherein the configuration manager comprises at
11 least one of:

12 means for managing a number of maximum allowable parallel data enhancement
13 processes,

14 means for managing types of single-source cleansing processes applied during a data
15 enhancement process,

16 means for managing types of cross-source processes applied during a data
17 enhancement process,

18 means for managing rules to be applied during specific single-source cleansing
19 processes, and

20 means for managing rules to be applied during specific cross-source processes.

21
22 19. The reference data utility of claim 1, wherein said output generator comprises:

23 means for receiving at least one request from a recipient;

24 means for parsing the at least one request to extract a request specification; and

25 means for initiating at least one work flow to provide the output data to the recipient.

26
27 20. A method for operating a reference data utility for serving a plurality of recipients,
28 comprising:

29 receiving unprocessed reference data inputs from a plurality of sources;

30 processing the unprocessed reference data received so as to generate processed
31 reference data having an increased value;

32 storing the unprocessed reference data and the processed reference data; and

1 generating output data for specified recipients; so that said output data contains only
2 at least one of unprocessed reference data and processed reference data, that said recipient is
3 entitled to receive.
4

5 21. The method of claim 20, further comprising configuring the reference data utility so as to
6 be scalable with respect to support for at least one of an increasing number of sources, an
7 increasing number of recipients, an increasing number of processes, and an increasing
8 number and complexity of entitlement arrangements.
9

10 22. The method of claim 20, further comprising storing a plurality of business documents the
11 repository, and generating as output a selected group of said documents.
12

13 23. The method of claim 20, further comprising storing access rights to sources, wherein the
14 data that a recipient is entitled to receive is defined by said access rights.
15

16 24. The method of claim 20, wherein the recipients are individuals granted entitlement to
17 particular sources of reference data and enhancement processes by at least one of a plurality
18 of tenant organizations sharing use of the reference data utility, said at least one of said tenant
19 organizations arranging independently with one or more data sources to have entitlements to
20 their data, and with the reference data utility to have entitlement to the results of applying
21 specific data enhancement processes to other reference data entitled to said at least one tenant
22 organization.
23

24 25. The method of claim 20, wherein the unprocessed reference data comprises information
25 elements, and the reference data utility annotates a plurality of said information elements with
26 sourcing information.
27

28 26. The method of claim 25, wherein the information elements have attributes, and the
29 reference data utility annotates said attributes with sourcing information.
30

31 27. The method of claim 26, further comprising maintaining information about entitlement of
32 recipients to said information elements, based on said sourcing information.
33

1 28. The method of claim 20, further comprising utilizing apparatus located in geographically
2 dispersed regions.

3
4 29. The method of claim 28, further comprising operating as an independent reference data
5 utility components located in one of said geographically dispersed regions.

6
7 30. The method of claim 29, wherein each independent reference data utility includes a local
8 repository, further comprising communicating information between said local repositories.

9
10 31. The method of claim 19, wherein each independent reference data utility is specialized to
11 provide information pertaining to a particular geographic region, further comprising
12 communicating information from other independent reference data utilities in other
13 geographic regions.

14
15 32. The method of claim 20, further comprising assessing accuracy of a source by a
16 combination of recording quality enhancement actions on values received from a source; and
17 comparing newly-arriving reference values with current multi-source recommended value for
18 that item; and recording the consistency with which a value provided from a source matches a
19 recommended value.

20
21 33. The method of claim 20, further comprising managing configuration of reference data
22 utility by managing at least one of:

23 a number of maximum allowable parallel data enhancement processes,
24 types of single-source cleansing processes applied during a data enhancement process,
25 types of cross-source processes applied during a data enhancement process,
26 rules to be applied during specific single-source cleansing processes, and
27 rules to be applied during specific cross-source processes.

28
29 34. The method of claim 20, wherein said generating output comprises:

30 receiving at least one request from a recipient;
31 parsing the at least one request to extract a request specification;
32 initiating at least one work flow to provide the output data to the recipient.

1 35. The method of claim 20, comprising providing value added services including at least one
2 service selected from the group consisting of data-driven value added computational
3 functions based on dynamically delivered input datasets, storage and retrieval of business
4 documents, rule-based validation of the applicability of stored business documents to a
5 business transaction and choreography of reference data associated with a business document
6 in support of a business transaction.

7
8 36. The method of claim 20, further comprising maintaining chronological accuracy within
9 the data flow across components of the reference data utility.

10
11 37. The method of claim 20, further comprising maintaining a record of total usages by
12 source for each recipient.

13
14 38. The method of claim 37, further comprising generating a report on at least one of source
15 usage and quality of source for each recipient.

16
17 39. The method of claim 20, further comprising creating a market for value added
18 computational services by:

19 establishing a registry for the available services;
20 accepting requests from recipients to execute an identified service with input data
21 provided an on demand dataset;
22 invoking the requested service;
23 returning results from the service computation to the requesting recipient using an on
24 demand dataset; and
25 monitoring service instances to record reporting information.

26
27 40. The method of claim 39, wherein the establishing a registry of available services
28 comprises:

29 providing a description of the service based on information from a service source, a
30 specification of reference data inputs required to use the service, specification of the outputs
31 generated by each service computation, and
32 maintaining entitlement information from the service origin identifying recipients
33 entitled to use the service.

1 41. The method of claim 20, further comprising handling recipient requests for an added
2 value service instance by receiving identification of requested service, specification of input
3 reference data used with the service, and delivery specification indicating how output from
4 the service is returned to a client.

5
6 42. The method of claim 41, wherein invoking a requested service comprises:
7 validating recipient entitlement to use the service;
8 collecting recipient specified input data by forming and executing an on-demand
9 dataset request to a delivery subsystem based on a transformation of the original request for
10 service execution;
11 verifying that recipient input data meets service input requirements; and
12 executing a service instance.

13
14 43. The method of claim 20, further comprising storing business documents with annotations
15 relating their content to reference data values.

16
17 44. The method of claim 20, further comprising accepting documents from at least one
18 recipient with reference data annotations, storing annotated documents in the repository, and
19 provide services to recipients based on information arriving from a source relating to said
20 annotations.

21
22 45. The method of claim 20, further comprising performing a validation test on current values
23 of at least one of unprocessed reference data and processed reference data.

24
25 46. A computer usable medium having computer readable program code means embodied
26 therein, the computer readable program code means being for causing a computer to effect
27 the method of any one of claims 20 to 46.

28
29
30 47. A multi-source multi-tenant data repository of data useful to requestors, comprising:
31 a database management system, said system managing a data base including
32 information elements from a plurality of sources, describing at least one referred entity;

1 annotations of sourcing information for said elements, and entitlements of requesters to
2 information elements based on said sourcing information;
3 data input apparatus for placing said information elements in said database; and
4 data output apparatus for providing data output based on said information elements in
5 response to a request from a requestor, said output containing information to which said
6 requestor is entitled based on said entitlements.

7
8 48. A repository as recited in claim 47, wherein the requestors are selected from the group
9 comprising: tenants, clients, sources of reference data, an agent acting on behalf of the
10 repository; software programs acting on behalf of a tenant; and software programs acting on
11 behalf of the repository.

12
13 49. A repository as recited in claim 47, further comprising means for authenticating a
14 requestor as authorized to obtain data from said repository.

15
16 50. A repository as recited in claim 47, wherein said entitlements comprise at least one of:
17 entitlements to data provided by a specific data source;
18 entitlements to data items generated by a particular value enhancement process; and
19 entitlements to data from a particular process, if all values used to derive the data
20 were from data sources to which a requestor is entitled.

21
22 51. A repository as recited in claim 50, comprising a database portion for storing
23 information concerning a set of entitlements governing outputs which a requestor associated
24 with a tenant of the repository is entitled to receive, based on tenants contracting
25 independently with data sources.

26
27 52. A repository as recited in claim 47, wherein said information elements comprise element
28 types taken from a list of element types comprising: item instances; versioned attributes;
29 properties; attribute values; business documents; function definitions; rule sets; operational
30 information; media documents; descriptions of standards; hierarchical business data; news
31 information; structured documents; configuration information; and program code.

1 53. A repository as recited in claim 47, wherein a repository entity comprises at least one
2 item instance information element, each item instance element comprising at least one
3 versioned attribute information element.
4

5 54. A repository as recited in claim 53, wherein a repository entity has at least one property
6 used to select said repository entity in information element selection operations.
7

8 55. A repository as recited in claim 53, wherein an item instance comprises at least one
9 property used to select said item instance in information element selection operations.
10

11 56. A repository as recited in claim 47, wherein said repository receives from at least one
12 item instance, process item instances or metadata for inclusion in the repository, wherein said
13 item instance process is uniquely identified by the repository; said repository further
14 comprising:

15 means for forming at least one repository entity corresponding to at least one referred
16 entity; and

17 means for associating said item instances or metadata for inclusion with at least one of
18 said repository entities, based on reference to said at least one referred entity.
19

20 57. A repository as recited in claim 47, wherein said database management system
21 comprises:

22 means for storing in said database as at least one property of at least one item
23 instance, information identifying an item instance process yielding said item instance;

24 means for storing in said database in the form of at least one evolutionarily tracked
25 source data tag at least one event in a history of said item instance, said at least one event
26 comprised of temporal information, sources and agents of said at least one event as received
27 from at least one item instance process;

28 means for storing in the form of at least one evolutionarily tracked source data tag
29 event in a history of versioned attributes of said item instance, said at least one event
30 comprised of temporal information, sources and agents of said at least one event as received
31 from an item instance process;

32 means for storing metadata for each said item instance;

33 means for storing metadata for each repository entity; and

1 means for storing metadata for each said versioned attribute.

2
3 58. A repository as recited in claim 47, wherein said database management system comprises
4 at least one of:

5 means for storing in at least one evolutionarily tracked source data tag at least one
6 event reflecting operations involving item instances, said at least one event comprised of
7 temporal information, sources and agents of said at least one event;

8 means for creating new item instances;

9 means for affecting existing item instances;

10 means for providing composite processing of a plurality of item instances;

11 means for supplying an item instance with attribute values for a repository entity
12 received from a source dataset;

13 means for supplying an item instance with attribute values for a repository entity
14 resulting from quality assurance and enhancement of values received from a single source
15 dataset; and

16 means for supplying an item instance with attribute values for a repository entity
17 resulting from comparison and selection between values from a plurality of source datasets
18 relating to the same attribute of the same referred entity.

19
20 59. A repository as recited in claim 47, wherein said database management system
21 comprises:

22 means for interpreting a request to produce at least one request specification;

23 means for selecting requested information elements in the repository based on said
24 request specification to form a return dataset;

25 means for filtering said return dataset to include only the information elements to
26 which said requester is entitled to form a filtered return dataset; and

27 means for supplying said filtered return dataset to said data output apparatus.

28
29 60. A repository as recited in claim 59, wherein the means for interpreting yields a request
30 specification comprising a plurality of aspects taken from a group of aspects comprising:
31 identification of the requester; a selection predicate governing selection of the information
32 elements to be returned; and a sourcing preference comprising prioritization of said at least

one preference of a requester when multiple information elements from separate sources are available satisfying said information element selection predicate.

61. A repository as recited in claim 58, wherein the means for selecting comprises at least one taken from a group comprising:

means for selecting any repository entity matching said request specification;

means for selecting any item instance belonging to said repository entities matching said request specification;

means for selecting any attribute value forming said item instances matching said request specification;

means for selecting any metadata associated with said any repository entity matching said request specification;

means for selecting any metadata associated with said any item instance matching said request specification;

means for selecting any metadata associated with said any attribute value matching said request specification; and

means for applying specified requester sourcing preference to form a return dataset comprising: said item instances, said attribute values belonging to said item instances, and said metadata.

62. A repository as recited in claim 58, wherein the means for filtering comprises at least one from a group comprising:

means for ensuring the entitlement of said at least one requester to any item instance process responsible for generating any of the item instances selected;

means for ensuring the entitlement of said at least one requester to any attribute value selected, wherein said requester is entitled to receive said any attribute value only if said requester is entitled to at least one source providing said attribute value; and

means for removing from the return dataset any information element to which the requester is not entitled.

63. An information processing apparatus including means for maintaining a multi-source multi-tenant data repository, wherein the means for maintaining comprises:

1 means for forming said multi-source multi-tenant data repository to include
2 information elements from a plurality of sources, describing at least one referred entity;
3 means for annotating a plurality of elements from said information elements in the
4 multi-source multi-tenant data repository with sourcing information; and
5 means for maintaining information about entitlement of requesters to information
6 elements based on said sourcing information.

7
8 64. A multi-source multi-tenant reference data repository, comprising:

9 means for inputting into said repository information about business instrument
10 information, corporate hierarchy information, corporate news information and other
11 information from multiple sources;

12 means for annotating changes to said business instrument information, corporate
13 hierarchy information, corporate news information, and other information, and sources of
14 each change;

15 means for maintaining information about entitlement of requesters to obtain said
16 business instrument information, corporate hierarchy information, corporate news
17 information, and other information, based on entitlement to the sources and processes which
18 have provided or modified this information; and

19 means for responding to at least one request from at least one requester to return at
20 least a subset of said business instrument information, corporate hierarchy information,
21 corporate news information, and other information, based on requester-specified selection
22 and sourcing preferences and subject to said entitlements.

23
24 65. A repository apparatus comprising:

25 means for forming the repository with information element structures in place;

26 means for inserting arriving information elements into a store of said the repository;

27 means for annotating each information element with annotations describing
28 evolutionary history of said each information element in forming an evolutionarily tracked
29 source data tag for said each information element;

30 means for maintaining source based entitlement information for authorized requesters
31 of repository information and data sources to which each authorized requester is entitled; and

1 means for employing said evolutionarily tracked source data tag together with source
2 based entitlements in a process providing entitlement enforced controlled access to
3 information included in the repository.
4
5

6 66. A method for maintaining a multi-source multi-tenant data repository of data useful to
7 requestors, comprising:

8 forming said multi-source multi-tenant data repository to include information
9 elements from a plurality of sources, describing at least one referred entity;
10 annotating a plurality of said elements with sourcing information; and
11 maintaining information about entitlement of requestors to information elements
12 based on said sourcing information.
13

14 67. A method as recited in claim 66, wherein said entitlements comprise at least one of:

15 entitlements to data provided by a specific data source;
16 entitlements to data items generated by a particular value enhancement process; and
17 entitlements to data from a particular process, if all values used to derive the data
18 were from data sources to which a requestor is entitled.
19

20 68. A method as recited in claim 67, wherein tenants of the repository contract
21 independently with data sources, and data enhancement process owners to determine a set of
22 entitlements governing data which requestors associated with tenants are entitled to receive.
23

24 69. A method as recited in claim 66, further comprising responding to at least one request
25 from at least one requester to return a set of information elements based on requester-
26 specified selection predicates and sourcing preferences and subject to said entitlement of said
27 at least one requester.
28

29 70. A method as recited in claim 66, wherein the step of responding comprises:

30 receiving at least one request from at least one requester;
31 interpreting said at least one request to produce at least one request specification;
32 selecting requested information elements in the multi-source multi-tenant data
33 repository based on said request specification to form a return dataset;

1 filtering said return dataset include only the information elements to which said
2 requester is entitled to form a filtered return dataset; and
3 returning said filtered return dataset to said requester.
4

5 71. A method as recited in claim 70, wherein the step of interpreting yields a request
6 specification comprising a plurality of aspects taken from a group of aspects comprising:
7 identification of the requester; a selection predicate governing selection of the information
8 elements to be returned; and a sourcing preference comprising prioritization of said at least
9 one requester's preference when multiple information elements from separate sources are
10 available satisfying said information element selection predicate.
11

12 72. A method as recited in claim 70, wherein the step of selecting comprises at least one step
13 taken from a group comprising:

14 selecting any repository entity matching said request specification;
15 selecting any item instance belonging to said repository entities matching said request
16 specification;

17 selecting any attribute value forming said item instances matching said request
18 specification;

19 selecting any metadata associated with said any repository entity matching said
20 request specification;

21 selecting any metadata associated with said any item instance matching said request
22 specification;

23 selecting any metadata associated with said any attribute value matching said request
24 specification; and

25 applying specified requester sourcing preference to form a return dataset comprising:
26 said item instances, said attribute values belonging to said item instances, and said metadata.
27

28 73. A method as recited in claim 70, wherein the step of filtering comprises at least one step
29 taken from a group of steps comprising:

30 ensuring the entitlement of said at least one requester to any item instance process
31 responsible for generating any of the item instances selected;

1 ensuring the entitlement of said at least one requester to any attribute value selected,
2 wherein said requester is entitled to receive said any attribute value only if said requester is
3 entitled to at least one source providing said attribute value; and

4 removing from the return dataset any information element to which the requester is
5 not entitled.

6
7 74. A method as recited in claim 66, wherein said multi-source multi-tenant data repository
8 comprises a plurality of information element types taken from a list of information element
9 types comprising: item instances; versioned attributes; properties; attribute values; business
10 documents; function definitions; rule sets; operational information; media documents;
11 descriptions of standards; hierarchical business data; news information; structured
12 documents; configuration information; and program code.

13
14 75. A method as recited in claim 66, wherein a repository entity has at least one item
15 instance information element, each item instance comprising at least one versioned attribute
16 information element.

17
18 76. A method as recited in claim 75, wherein a repository entity comprises at least one
19 property used to select said repository entity in information element selection operations.

20
21 77. A method as recited in claim 75 wherein an item instance comprises at least one property
22 used to select said item instance in information element selection operations.

23
24 78. A method as recited in claim 66, wherein the forming comprises at least one step taken
25 from a group of steps comprising:
26 said multi-source multi-tenant data repository receiving from at least one item
27 instance process item instances or metadata for inclusion in the repository, wherein said item
28 instance process is uniquely identified by the repository;
29 said repository forming at least one repository entity corresponding to at least one
30 referred entity; and
31 said repository associating said item instances or metadata for inclusion with at least
32 one of said repository entities based on reference to said at least one referred entity.

1 79. A method as recited in claim 66, wherein the annotating comprises at least one step taken
2 from a group of steps comprising:

3 storing as at least one property of at least one item instance, information identifying-
4 an item instance process yielding said item instance;

5 storing in the form of at least one evolutionarily tracked source data tag at least one
6 event in a history of said item instance, said at least one event comprised of temporal
7 information, sources and agents of said at least one event as received from at least one item
8 instance process;

9 storing in the form of at least one evolutionarily tracked source data tag event in a
10 history of versioned attributes of said item instance, said at least one event comprised of
11 temporal information, sources and agents of said at least one event as received from an item
12 instance process;

13 storing metadata for each said item instance;

14 storing metadata for each repository entity; and

15 storing metadata for each said versioned attribute.
16

17 80. A method as recited in claim 78, wherein said repository item instance process comprises
18 at least one step taken from a group of steps comprising:

19 storing in at least one evolutionarily tracked source data tag at least one event
20 reflecting operations involving item instances, said at least one event comprised of temporal
21 information, sources and agents of said at least one event;

22 employing a process that creates new item instances;

23 employing a process that affects existing item instances;

24 employing a composite process comprised of a plurality of item instance processes;

25 supplying an item instance with attribute values for a repository entity received from a
26 source dataset;

27 supplying an item instance with attribute values for a repository entity resulting from
28 quality assurance and enhancement of values received from a single source dataset; and

29 supplying an item instance with attribute values for a repository entity resulting from
30 comparison and selection between values from a plurality of source datasets relating to the
31 same attribute of the same referred entity.
32

1 81. A method as recited in claim 66, wherein said requester is taken from a group of
2 requesters comprising: an agent acting on behalf of a repository tenant, an agent acting on
3 behalf of the repository; software programs acting on behalf of a repository tenant; and
4 software programs acting on behalf of the repository.

5
6 82. A method for maintaining a multi-source multi-tenant reference data repository,
7 comprising:

8 forming said multi-source multi-tenant reference data repository to include
9 information about business instrument information, corporate hierarchy information,
10 corporate news information and other information from multiple sources;

11 annotating changes to said business instrument information, corporate hierarchy
12 information, corporate news information, and other information, and sources of each change;

13 maintaining information about entitlement of requesters to obtain said business
14 instrument information, corporate hierarchy information, corporate news information, and
15 other information, based on entitlement to the sources and processes which have provided or
16 modified this information; and

17 responding to at least one request from at least one requester to return at least a subset
18 of said business instrument information, corporate hierarchy information, corporate news
19 information, and other information, based on requester-specified selection and sourcing
20 preferences and subject to said entitlements.

21
22 83. A method as recited in claim 82, wherein said sources are from a group of sources
23 comprising: providers of business data; providers of financial data; providers of ratings data;
24 providers of corporate news data; and providers of corporate hierarchy data.

25
26 84. A method as recited in claim 82, wherein said information includes financial information
27 data taken from a group of data comprising at least one of: stock instruments; bond
28 instruments; derivative instruments; fixed income instruments; mortgage-backed instruments;
29 counterparty information; financial transaction information; corporate events information;
30 financial trade information; financial settlement information; financial documents; other
31 financial instruments; and other financial data.

1 85. A method as recited in claim 82, wherein the forming comprises keeping distinct versions
2 of historical states of said business instrument information, corporate hierarchy information,
3 corporate news information and other information.
4

5 86. A data repository method comprising:

6 forming the repository with information element structures in place;
7 inserting arriving information elements into a store of said the repository;
8 annotating each information element with annotations describing evolutionary history
9 of said each information element in forming an evolutionarily tracked source data tag for said
10 each information element;

11 maintaining source based entitlement information for authorized requesters of
12 repository information and data sources to which each authorized requester is entitled; and

13 employing said evolutionarily tracked source data tag together with source based
14 entitlements in a process providing entitlement enforced controlled access to information
15 included in the repository.
16

17 87. A method as recited in claim 86, further comprising retrieval requests arriving as input;
18 and the process of providing comprising forming retrieval responses being returned as output
19 for this processing.
20

21 88. A method as recited in claim 87, further comprising employing the repository to store
22 other items that reside in a data store, said other items comprising at least one of: business
23 value added functions, business documents, functional rule sets; operational rule sets,
24 functional log records, and operational log records.
25

26 89. A method as recited in claim 86, further comprising associating the evolutionarily
27 tracked source data tags with at least one information element in the repository; and the step
28 annotating comprises recording sourcing information to provide traceability to sources that
29 contributed to creation of an information element value.
30

31 90. A computer usable medium having computer readable program code means embodied
32 therein, the computer readable program code means being for causing a computer to effect
33 the method of any one of claims 66 to 89.

1 91. A method for enhancing the value of reference data, comprising:

2 subjecting the data to at least one value enhancing process; and
3 maintaining a complete record of all sources of the data and all enhancement
4 processing steps contributing to the generation of each enhanced element of the reference
5 data.

6
7 92. A method as recited in claim 91, further comprising:

8 receiving data concerning a referred item from a first data source; and
9 generating enhanced values based on comparing and processing values for
10 the same referred item from multiple sources.

11
12 93. A method as recited in claim 91, further comprising performing at least one of:

13 validating the data by at least one of a manual process and an automatic process;
14 normalizing the data by at least one of a manual process and an automatic process;
15 and
16 cleansing the data by at least one of a manual process and an automatic process.

17
18 94. A method as recited in claim 93, wherein said reference data includes source elements,
19 and said validating comprises:

20 obtaining said at least one source element from a source description; and
21 performing at least one step taken from a group of steps comprising:
22 detecting any source element which does not conform to the source
23 description;
24 flagging any source element which does not conform to the source description;
25 correcting any source element which does not conform to the source
26 description; and
27 removing any source element which does not conform to the source
28 description; and

29
30 recording to at least one evolutionarily tracked sourced data tag any event generated
31 by said step of performing validation.

1 95. A method as recited in claim 93, wherein said reference data includes source elements, and said
2 normalizing comprises:
3 obtaining said source element in a source description;
4 converting said source element based on said source description to at least one target
5 information element based on a corresponding target description, wherein the target description is
6 information describing structure, contents and constraints of repository information elements, as they
7 are stored in a repository; and
8 performing at least one step taken from a group of steps comprising:
9 detecting any source element which cannot be normalized;
10 flagging any source element which cannot be normalized;
11 correcting any source element which cannot be normalized;
12 removing any source element which cannot be normalized; and
13 recording to at least one evolutionarily tracked sourced data tag any event generated
14 by said step of performing normalization.

15
16 96. A method as recited in claim 93, wherein said reference data includes source elements,
17 and said cleansing comprises at least one of:
18 automated execution of at least one rule from at least one rule set containing source-
19 specific cleansing rules;
20 examination of said source element values by one skilled in subject matter relevant to
21 at least one referred entity;
22 application of any rule from said at least one rule set containing source-specific rules
23 by one skilled in subject matter relevant to at least one referred entity;
24 removal of any of said source element values;
25 augmentation of any of said source element values;
26 correction of any of said source element values;
27 annotation of any quality concerns;
28 reporting back to the source, inquiries regarding quality of the source element in
29 question; and
30 recording any event generated by any action, taken from said group of actions, to at
31 least one evolutionarily tracked sourced data tag.

1 97. A method as recited in claim 91, further comprising receiving said reference data from
2 multiple sources, and selecting and enhancing the data by at least one of a manual process
3 and an automatic process to produce data of enhanced value.
4

5 98. A method as recited in claim 97, comprising:

6 selecting all of the source elements that contain information describing a same
7 referred entity;

8 applying predetermined rules to at least one of the source elements and attributes of
9 the elements;

10 selecting one of a preferred or recommended item from the alternatives provided by
11 the different sources by at least one of:

12 creating at least one new item based on a combination of attributes provided by
13 the different sources; or

14 modifying the elements provided by the different sources;

15 creating a new corresponding evolutionarily tracked source data tag when at least one
16 new item or items is created;

17 annotating said evolutionarily tracked source data tag at the source item level with the
18 information about the cross-source processing applied to the item.
19

20 99. A method as recited in claim 98, wherein if an existing element was selected but no
21 attributes were modified, the method further comprises providing an annotation at the item
22 level to denote which parent sources matched the selection made.
23

24 100. A method as recited in claim 98, wherein if either modification of data at an attribute
25 level or a creation of a new item occurs, the method further comprises separately annotating
26 an exact set of sources for each attribute.
27

28 101. A data processing method comprising producing at least one evolutionarily tracked
29 source tagged dataset, comprising:

30 receiving at least one source-dataset from at least one source, wherein a source
31 element includes one of a source attribute and a source item, each source-dataset having at
32 least one source item, each source item having at least one source attribute;

1 recording a source identification for each source element, and a source identification
2 for each source-dataset in at least one evolutionarily tracked source data tag;
3 obtaining relevant information resulting from the step of receiving and the step of
4 recording to form at least one recordable event in at least one evolutionarily tracked source
5 data tag; and
6 forming said at least one evolutionarily tracked source tagged dataset to include at
7 least one evolutionarily tracked source data tag, said at least one evolutionarily tracked source
8 data tag including said at least one recordable event, and including at least one source of said
9 at least one recordable event.

10
11 102. A method as recited in claim 101, further comprising:

12 invoking at least one rule from at least one rule-set on at least one of said source
13 dataset, said source element, and an information element; and
14 obtaining relevant information evolving from the step of invoking to form at least one
15 other recordable event in at least one evolutionarily tracked source data tag.

16
17 103. A method as recited in claim 102, wherein said at least one rule set comprises at least
18 one rule taken from a group of rules, comprising:

19 rules for checking range tolerance of source attribute values;
20 rules for checking rate of change of source attribute values;
21 rules for checking consistency of source attribute values with other relevant source
22 attribute values;
23 rules for checking structural consistency of source elements;
24 rules for checking consistency of source elements with other relevant source
25 elements;
26 rules for checking suitability of source elements for transformation into target
27 information elements within a multi-source multi-tenant data repository, as described by a
28 target description;
29 rules for checking compatibility of source element values with existing referred entity
30 information;
31 rules for identifying source elements as having come from a particular source;
32
33 rules for comparing source elements in the context of a specific cross-source process;

1 rules applicable to source datasets;
2 rules applicable to source elements; and
3 rules applicable to information elements.
4

5 104. A method as recited in claim 103, wherein said at least one rule is grouped into at least
6 one rule set according to applicability of said at least one rule to at least one processing stage
7 taken from a group of processing stages, comprising: validation; normalization; source-
8 specific cleansing; and a cross-source process.
9

10 105. A method as recited in claim 102, wherein a rule comprises at least one of: an
11 executable test condition; a correction method; information identifying said at least one rule
12 set to which said rule belongs.
13

14 106. A method as recited in claim 102, wherein a recordable event includes data taken from
15 a group of data comprising: an event description; an agent of the event; temporal information
16 associated with the event; at least one source of the event; an identifier of the event;
17 information required to correlate the event with the information element to which it applies;
18 and a classification of the event.
19

20 107. A method as recited in claim 102, wherein the step of invoking comprises at least one
21 step taken from a group of steps comprising: performing validation on at least one source
22 element; performing normalization on said at least one source element; performing source-
23 specific cleansing on said at least one source element; and executing at least one cross-source
24 process on said at least one source element.
25

26 108. A method as recited in claim 107, wherein the step of performing validation on said at
27 least one source element comprises:

28 obtaining said at least one source element from a source description; and
29 performing at least one step taken from a group of steps comprising:
30 detecting any source element which does not conform to the source
31 description;
32 flagging any source element which does not conform to the source description;

1 correcting any source element which does not conform to the source
2 description; and

3 removing any source element which does not conform to the source
4 description; and

5 recording to at least one evolutionarily tracked sourced data tag any event generated
6 by said step of performing validation.

7
8 109. A method as recited in claim 107, wherein the step of performing normalization on said
9 at least one source element comprises:

10 obtaining said source element in a source description;

11 converting said source element based on said source description to at least one target
12 information element based on a corresponding target description, wherein the target
13 description is information describing structure, contents and constraints of repository
14 information elements, as they are stored in a repository; and

15 performing at least one step taken from a group of steps comprising:

16 detecting any source element which cannot be normalized;

17 flagging any source element which cannot be normalized;

18 correcting any source element which cannot be normalized;

19 removing any source element which cannot be normalized; and

20 recording to at least one evolutionarily tracked sourced data tag any event generated
21 by said step of performing normalization.

22
23 110. A method as recited in claim 107, wherein the step of performing source-specific
24 cleansing comprises an action taken from a group of actions comprising:

25 automated execution of said at least one rule from said at least one rule set containing
26 source-specific cleansing rules;

27 examination of said source element values by one skilled in subject matter relevant to
28 at least one referred entity;

29 application of any rule from said at least one rule set containing source-specific rules
30 by one skilled in subject matter relevant to at least one referred entity;

31 removal of any of said source element values;

32 augmentation of any of said source element values;

33 correction of any of said source element values;

1 annotation of any quality concerns;
2 reporting back to the source, inquiries regarding quality of the source element in
3 question; and
4 recording any event generated by any action, taken from said group of actions, to at
5 least one evolutionarily tracked sourced data tag.

6

7 111. A method as recited in claim 107, wherein the step of executing at least one cross-source
8 process comprises an action taken from a group of actions comprising:

9 examining source elements from a plurality of data sources referring to a same
10 referred entity;

11 automatically executing at least one rule from said at least one rule set including
12 cross-source process rules specific to said at least one cross-source process;

13 examining said source elements by one skilled in subject matter relevant to said same
14 referred entity;

15 applying any rule from said at least one rule set containing cross-source process rules
16 specific to said at least one cross-source process by one skilled in such subject matter;

17 selecting any of said source elements values as a preferred value;

18 comparing any of said source elements;

19 removing any of said source element values;

20 augmenting any of said source element values;

21 modifying any of said source element values;

22 annotating any quality concerns;

23 creating at least one item instance to include results of said at least one cross-source
24 process;

25 modifying at least one item instance to include the results of said at least one cross-
26 source process;

27 adding identification information to at least one item instance to recognize said at
28 least one item instance as target of said at least one cross-source process; and

29 recording any event generated by any action, taken from said group of actions, to at
30 least one evolutionarily tracked sourced data tag.

31

1 112. A method as recited in claim 111, further comprising resolving differences detected
2 during the step of comparing said source elements through at least one step taken from a
3 group of steps comprising:

4 automatically selecting source elements based on business rules;
5 automatically selecting source elements based on algorithms;
6 manually selecting a recommended source element by one skilled in the subject,
7 based on knowledge of said subject area;
8 manually selecting a recommended source element by one skilled in the subject,
9 based on freely available public information;
10 manually creating a recommended source element by one skilled in the subject, based
11 on knowledge of the subject area;
12 manually creating a recommended source element by one skilled in the subject, based
13 on freely available public information; and
14 recording any event generated by any step taken from said group of steps, to at least
15 one evolutionarily tracked sourced data tag.

16
17 113. A method as recited in claim 111, wherein the step of recording comprises identifying
18 which sources matched a selected preferred source element value.

19
20 114. A method as recited in claim 108, further comprising:

21 presenting said at least one source element to one skilled in such subject;
22 enabling performance of manual validation of said at least one source element;
23 performing manual validation; and
24 recording to at least one evolutionarily tracked sourced data tag any event generated
25 by the step of performing manual normalization.

26
27 115. A method as recited in claim 109, further comprising:

28 presenting said at least one source element to one skilled in such subject;
29 enabling performance of manual normalization of said at least one source element;
30 performing manual normalization; and
31 recording to at least one evolutionarily tracked sourced data tag any event generated
32 by the step of performing manual normalization.

1 116. A method as recited in claim 101, wherein an overall set of reference data being
2 processed is on a variety of distinct topics, with the source datasets of reference data being
3 individually cleansed, each source supplying source items on at least one topic.
4

5 117. A data processing method for quality assurance of reference data, comprising:
6 receiving reference data in a source dataset from at least one source, each source-
7 dataset having at least one source item, each source item having at least one source attribute,
8 wherein a source element is one of a source item and a source attribute;

9 recording a source identification for each source element, and a source identification
10 for each source-dataset in at least one evolutionarily tracked source data tag, such that at least
11 one evolutionarily tracked source data tag is associated with each source element;

12 recording data evolution events from steps of validating, normalizing, single-source
13 processing, and cross-source processing, of source elements in said at least one evolutionarily
14 tracked source data tag; and

15 forming said at least one evolutionarily tracked source tagged dataset to include at
16 least one evolutionarily tracked source data tag, said at least one evolutionarily tracked source
17 data tag including said at least one data evolution event and a source of said at least one data
18 evolution event.
19

20 118. An article of manufacture comprising a computer usable medium having computer
21 readable program code means embodied therein for causing data processing, the computer
22 readable program code means in said article of manufacture comprising computer readable
23 program code means for causing a computer to effect the steps of any one of claims 91 to
24 117.
25

26 119. An apparatus for enhancing the value of reference data, comprising:
27 means for subjecting the data to at least one value enhancing process; and
28 a database for maintaining a complete record of all sources of the data and all
29 enhancement processing steps contributing to the generation of each enhanced element of the
30 reference data.
31

32 120. An apparatus as recited in claim 119, further comprising:
33 means for receiving data concerning a referred item from a first data source; and

means for generating enhanced values based on comparing and processing values for the same referred item from multiple sources.

121. An apparatus as recited in claim 119, further comprising at least one of:

validating means for validating the data by at least one of a manual process and an automatic process;

normalizing means for normalizing the data by at least one of a manual process and an automatic process; and

cleansing means for cleansing the data by at least one of a manual process and an automatic process.

122. An apparatus as recited in claim 121, wherein said reference data includes source elements, and said validating means comprises:

means for obtaining said at least one source element from a source description; and

means for performing at least one step taken from a group of steps comprising:

detecting any source element which does not conform to the source description;

flagging any source element which does not conform to the source description;

correcting any source element which does not conform to the source description; and

removing any source element which does not conform to the source description; and

means for recording to at least one evolutionarily tracked sourced data tag any event generated by said step of performing validation.

123. An apparatus as recited in claim 121, wherein said reference data includes source elements, and said means for normalizing comprises:

means for obtaining said source element in a source description;

means for converting said source element based on said source description to at least one target information element based on a corresponding target description, wherein the target description is information describing structure, contents and constraints of repository information elements, as they are stored in a repository; and

means for performing at least one step taken from a group of steps comprising:

1 detecting any source element which cannot be normalized;
2 flagging any source element which cannot be normalized;
3 correcting any source element which cannot be normalized;
4 means for removing any source element which cannot be normalized; and
5 means for recording to at least one evolutionarily tracked sourced data tag any event
6 generated by said step of performing normalization.
7

8 124. An apparatus as recited in claim 121, wherein said reference data includes source
9 elements, and said cleansing means comprises at least one of:

10 means for automated execution of at least one rule from at least one rule set
11 containing source-specific cleansing rules;
12 means for examination of said source element values by one skilled in subject matter
13 relevant to at least one referred entity;
14 means for application of any rule from said at least one rule set containing source-
15 specific rules by one skilled in subject matter relevant to at least one referred entity;
16 means for removal of any of said source element values;
17 means for augmentation of any of said source element values;
18 means for correction of any of said source element values;
19 means for annotation of any quality concerns;
20 means for reporting back to the source, inquiries regarding quality of the source
21 element in question; and
22 means for recording any event generated by any action, taken from said group of
23 actions, to at least one evolutionarily tracked sourced data tag.
24

25 125. An apparatus as recited in claim 119, further comprising means for receiving said
26 reference data from multiple sources, and means for selecting and enhancing the data by at
27 least one of a manual process and an automatic process to produce data of enhanced value.
28

29 126. An apparatus as recited in claim 125, comprising:

30 means for selecting all of the source elements that contain information describing a
31 same referred entity;
32 means for applying predetermined rules to at least one of the source elements and
33 attributes of the elements;

1 means for selecting one of a preferred or recommended item from the alternatives
2 provided by the different sources by at least one of:

3 creating at least one new item based on a combination of attributes provided by
4 the different sources; or

5 modifying the elements provided by the different sources;

6 means for creating a new corresponding evolutionarily tracked source data tag when
7 at least one new item or items is created; and

8 means for annotating said evolutionarily tracked source data tag at the source item
9 level with the information about the cross-source processing applied to the item.

10
11 127. An apparatus as recited in claim 126, further comprising means for providing an
12 annotation at the item level to denote which parent sources matched the selection made, if an
13 existing element was selected but no attributes were modified.

14
15 128. An apparatus as recited in claim 126, further comprising means for separately annotating
16 an exact set of sources for each attribute, if either modification of data at an attribute level or
17 a creation of a new item occurs.

18
19 129. A data processing apparatus for producing at least one evolutionarily tracked source
20 tagged dataset, comprising:

21 at least one input for receiving at least one source-dataset from at least one source,
22 each source-dataset having at least one source item, each source item having at least one
23 source attribute;

24 memory for recording a source identification for each source attribute, a source
25 identification for each source item, and a source identification for each source-dataset;

26 apparatus for invoking at least one rule from at least one rule-set on at least one of:

27 said source-dataset;

28 said source item; and

29 said attribute; and

30 apparatus for retaining relevant information about the steps of invoking, receiving and
31 recording resulting in at least one recordable event; and

1 a processor for forming said at least one evolutionarily tracked source tagged dataset
2 to include said at least one recordable event and an event originator of said at least one
3 recordable event.
4

5 130. A data processing apparatus for assuring quality of reference data, comprising:

6 means for receiving reference data in a source dataset from at least one source, each
7 source-dataset having at least one source item, each source item having at least one source
8 attribute, wherein a source element is one of a source item and a source attribute;

9 means for recording a source identification for each source element, and a source
10 identification for each source-dataset in at least one evolutionarily tracked source data tag,
11 such that at least one evolutionarily tracked source data tag is associated with each source
12 element;

13 means for recording data evolution events from steps of validating, normalizing,
14 single-source processing, and cross-source processing, of source elements in said at least one
15 evolutionarily tracked source data tag; and

16 means for forming said at least one evolutionarily tracked source tagged dataset to
17 include at least one evolutionarily tracked source data tag, said at least one evolutionarily
18 tracked source data tag including said at least one data evolution event and a source of said at
19 least one data evolution event.
20
21

22 131. An information delivery method for satisfying at least one on-demand dataset request,
23 comprising:

24 processing said at least one on-demand dataset request from at least one requester;

25 producing at least one parsed on-demand dataset request specification;

26 configuring at least one on demand dataset production process to produce at least one
27 on-demand dataset satisfying said at least one on-demand dataset request; and

28 executing said at least one on demand dataset production process to return said at least
29 one on-demand dataset to said at least one requester;

30 wherein said on-demand dataset is limited to data derived from sources and data
31 enhancements to which said requester is entitled.
32

1 132. A method as recited in claim 131, wherein said on demand dataset request comprises at
2 least one on demand dataset request specification enabling said requester to specify
3 characteristics taken from a group of characteristics comprising:

4 information items to be returned; selection of information items to be returned;
5 sourcing preference to select between alternate available values; delivery mode; delivery
6 timing; transport protocol; transport protocol ports; security tokens; preferred data format;
7 data transformation rules; custom functions to be invoked; custom filtering rules; exception
8 handling instructions; annotation instructions; data delivery feedback mechanism
9 instructions; delivery endpoint; delivery intermediaries; metadata handling instructions;
10 logging instructions; routing instructions; data merging instructions; and data splitting
11 instructions.

12
13 133. A method as recited in claim 131, further comprising receiving information assembled
14 in an on demand dataset from a multi-source multi-tenant data repository.

15
16 134. A method as recited in claim 131, wherein said at least one requester is taken from a
17 group of requesters comprising:

18 a tenant of a multi-source multi-tenant repository;
19 an agent acting on behalf of said tenant;
20 an agent acting on behalf of said repository;
21 a program acting on behalf of said repository; and
22 a program acting on behalf of said tenant.

23
24 135. A method as recited in claim 131, further comprising using a delivery mode to deliver
25 said at least one on demand dataset, said delivery mode including at least one delivery
26 modality from a group of delivery modalities comprising:

27 quasi real time delivery; scheduled batch delivery; datamart delivery; one time query
28 delivery; e-mail delivery; fax delivery; online delivery; printed hard copy delivery; automated
29 voice delivery; magnetic tape delivery; optical disc delivery; digital media delivery; video
30 delivery; and condition triggered delivery.

1 136. A method as recited in claim 131, wherein the configuring said at least one on demand
2 dataset production process is enabled by employing at least one activity building block for
3 separable steps of said at least one on demand dataset production process.

4
5 137. A method as recited in claim 136, wherein said at least one activity building block is
6 taken from a collection of activity building blocks wherein each block enables at least one
7 function taken from a group of functions comprising:

8 information element selection; sourcing selection; entitlement enforcement; data
9 assembly; delivery scheduling; transfer protocol handling; standard format transformation;
10 database load; custom data transformation; logging; function execution; filtering; annotating;
11 routing; data splitting; data merging; and data transmission.

12
13 138. A method as recited in claim 131, wherein said step of configuring at least one on
14 demand dataset production process comprises:

15 employing said at least one parsed on demand dataset request specification;
16 selecting at least one activity building block for inclusion in the on demand dataset
17 production process, wherein said at least one activity building block satisfies at least one
18 characteristic of said at least one parsed on demand dataset request specification;
19 parameterizing any selected activity building block with execution parameters; and
20 assembling selected parameterized activity blocks into said at least one on demand
21 dataset production process.

22
23 139. A method as recited in claim 131, wherein the executing said on demand dataset
24 production process includes at least one step taken from a group of steps comprising:

25 executing logic of an assembly flow included within said on demand dataset
26 production process;
27 executing each parameterized activity building block of said on demand dataset
28 production process as many times as indicated by said logic;
29 transmitting said at least one on demand dataset to said at least one requester;
30 recording aspects of actions taken in response to said at least one request to enable
31 repetition of said step of executing at a later time; and
32 logging at least one aspect of delivery taken from a group of aspects including:

1 time of delivery; date of delivery; contents of delivery; requester of the delivery;
2 mode of the delivery; size of the delivery; execution time of the delivery process; identifier of
3 the delivery; any error of the delivery process; any warning of the delivery process; success
4 of the delivery process; feedback of the delivery process; non-repudiation information
5 associated with the delivery process; security features of the delivery; and said at least one on
6 demand dataset request.

7
8 140. A method as recited in claim 131, wherein said on demand dataset request is initiated by
9 an action taken from a group of actions comprising: a manually initiated request; an
10 automatically initiated request; a one-time request; a data arrival event; a data availability
11 event; a data deletion event; a data change event; a data temporal event; a scheduled request;
12 a request received through an intermediary; and an on-line initiated request.

13
14 141. A method as recited in claim 131, the method being scalable to permit delivery of
15 information via multiple delivery requests from multiple requesters, the method utilizing
16 automated handling of information delivery requests, the method being specifically
17 configured for needs of each delivery request.

18
19 142. A method for returning reference data from a multi-source multi-tenant data repository
20 in response to requests from requesters, comprising:

21 receiving at least one request from a requester;

22 parsing said at least one request to extract the request specification;

23 configuring at least one workflow to deliver the requested reference data based on
24 entitlements of said requester, as well as the selection criteria, sourcing preferences and other
25 preferences contained in requester's request; and

26 executing said workflow and delivering said requested reference data to the requester.

27
28 143. A method as recited in claim 142, wherein said request specification includes at least
29 one preference taken from a group of preferences comprising:

30 selection criteria; sourcing preferences; data format preferences; delivery transport
31 preferences; and preferences particular to said requester.

1 144. A method as recited in claim 142, wherein the configuring includes at least one action
2 taken from a group of actions comprising:

3 retrieving said requested reference data;
4 filtering said requested reference data; and
5 formatting said requested reference data.

6
7 145. An article of manufacture comprising a computer usable medium having computer
8 readable program code means embodied therein for causing information processing, the
9 computer readable program code means in said article of manufacture comprising computer
10 readable program code means for causing a computer to effect the steps of any one of claims
11 131 to 144.

12
13 146. An information processing apparatus for satisfying at least one on demand dataset
14 request, comprising:

15 a processor for processing said at least one on demand dataset request from at least
16 one requester;

17 a computer program component executable for producing at least one parsed on
18 demand dataset request specification;

19 a program configuration means for configuring at least one on demand dataset
20 production process to produce at least one on-demand dataset satisfying said at least one on
21 demand dataset request; and

22 computer code for executing said at least one on demand dataset production process
23 to return said at least one on-demand dataset to said at least one requester;

24 wherein said on-demand dataset is limited to data derived from sources and data
25 enhancements to which said requester is entitled.

26
27 147. An apparatus as recited in claim 146, wherein said processor processes an on demand
28 dataset request comprises at least one on demand dataset request specification enabling said
29 requester to specify characteristics taken from a group of characteristics comprising:

30 information items to be returned; selection of information items to be returned;
31 sourcing preference to select between alternate available values; delivery mode; delivery
32 timing; transport protocol; transport protocol ports; security tokens; preferred data format;
33 data transformation rules; custom functions to be invoked; custom filtering rules; exception

1 handling instructions; annotation instructions; data delivery feedback mechanism
2 instructions; delivery endpoint; delivery intermediaries; metadata handling instructions;
3 logging instructions; routing instructions; data merging instructions; and data splitting
4 instructions.

5
6 148. An apparatus as recited in claim 146, further comprising means for receiving
7 information assembled in an on demand dataset from a multi-source multi-tenant data
8 repository.

9
10 149. An apparatus as recited in claim 146, further comprising means for receiving said on
11 demand dataset request from at least one requester taken from a group of requesters
12 comprising:

13 a tenant of a multi-source multi-tenant repository;
14 an agent acting on behalf of said tenant;
15 an agent acting on behalf of said repository;
16 a program acting on behalf of said repository; and
17 a program acting on behalf of said tenant.

18
19 150. An apparatus as recited in claim 146, further comprising a delivery component, wherein
20 said delivery component including at least one delivery modality from a group of delivery
21 modalities comprising:

22 quasi real time delivery; scheduled batch delivery; datamart delivery; one time query
23 delivery; e-mail delivery; fax delivery; online delivery; printed hard copy delivery; automated
24 voice delivery; magnetic tape delivery; optical disc delivery; digital media delivery; video
25 delivery; and condition triggered delivery.

26
27 151. An apparatus as recited in claim 146, further comprising at least one activity building
28 block for building separable steps of said at least one on demand dataset production process
29 so as to configure said at least one on demand dataset production process.

30
31 152. An apparatus as recited in claim 151, wherein said at least one activity building block is
32 taken from a collection of activity building blocks wherein each block enables at least one
33 function taken from a group of functions comprising:

1 information element selection; sourcing selection; entitlement enforcement; data assembly;
2 delivery scheduling; transfer protocol handling; standard format transformation; database
3 load; custom data transformation; logging; function execution; filtering; annotating; routing;
4 data splitting; data merging; and data transmission.

5
6 153. An apparatus as recited in claim 146, wherein said program configuration means
7 comprises:

8 means for employing said at least one parsed on demand dataset request specification;
9 means for selecting at least one activity building block for inclusion in the on demand
10 dataset production process, wherein said at least one activity building block satisfies at least
11 one characteristic of said at least one parsed on demand dataset request specification;
12 means for parameterizing a selected activity building block with execution
13 parameters; and
14 means for assembling selected parameterized activity blocks into said at least one on
15 demand dataset production process.

16
17 154. An apparatus as recited in claim 146, wherein the computer code for executing said on
18 demand dataset production process includes at least one computer code component taken
19 from a group of computer code components, comprising:

20 computer code for executing logic of an assembly flow included within said on
21 demand dataset production process;
22 computer code for executing each parameterized activity building block of said on
23 demand dataset production process as many times as indicated by said logic;
24 computer code for transmitting said at least one on demand dataset to said at least one
25 requester;
26 computer code for recording aspects of actions taken in response to said at least one
27 request to enable repetition of said step of executing at a later time; and
28 computer code for logging at least one aspect of delivery taken from a group of
29 aspects including:
30 time of delivery; date of delivery; contents of delivery; requester of the delivery;
31 mode of the delivery; size of the delivery; execution time of the delivery process; identifier of
32 the delivery; any error of the delivery process; any warning of the delivery process; success
33 of the delivery process; feedback of the delivery process; non-repudiation information

1 associated with the delivery process; security features of the delivery; and said at least one on
2 demand dataset request.

3
4 155. An apparatus as recited in claim 146, further comprising means for initiating said on
5 demand dataset request in response to an action taken from a group of actions comprising:
6 a manually initiated request; an automatically initiated request; a one-time request; a
7 data arrival event; a data availability event; a data deletion event; a data change event; a data
8 temporal event; a scheduled request; a request received through an intermediary; and an on-
9 line initiated request.

10
11 156. An apparatus for returning reference data from a multi-source multi-tenant data
12 repository in response to requests from requesters, comprising:
13 means for receiving at least one request from a requester;
14 means for parsing said at least one request to extract the request specification;
15 means for configuring at least one workflow to deliver the requested reference data
16 based on entitlements of said requester, as well as the selection criteria, sourcing preferences
17 and other preferences contained in requester's request;
18 means for executing said workflow and delivering said requested reference data to the
19 requester; and
20 means for limiting said on-demand dataset to data derived from sources and data
21 enhancements to which said requester is entitled.

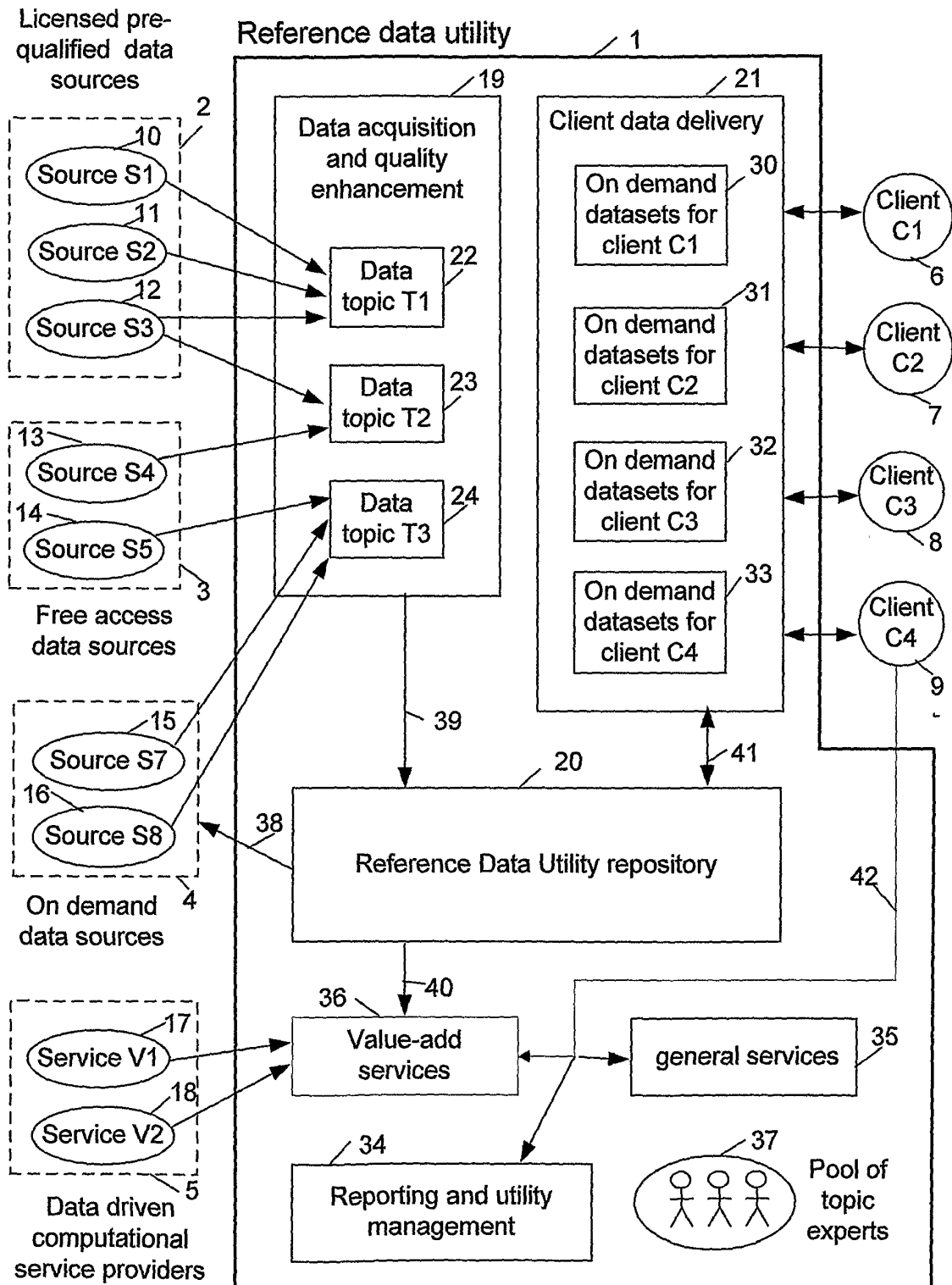
22
23 157. An apparatus as recited in claim 156, comprising means responsive to said request
24 specification including at least one preference taken from a group of preferences comprising:
25 selection criteria; sourcing preferences; data format preferences; delivery transport
26 preferences; and preferences particular to said requester.

27
28 158. An apparatus as recited in claim 156, wherein said means for configuring includes at
29 least one means taken from a group of means comprising:
30 means for retrieving said requested reference data;
31 means for filtering said requested reference data; and
32 means for formatting said requested reference data.
33

- 1 159. An apparatus for returning reference data from a multi-source multi-tenant data
2 repository in response to requests from requesters, comprising:
3 means for receiving at least one request from a requester;
4 means for parsing said at least one request to extract the request specification;
5 means for configuring at least one workflow to deliver the requested reference data
6 based on entitlements of said requester, as well as the selection criteria, sourcing preferences
7 and other preferences contained in requester's request; and
8 means for executing said workflow and delivering said requested reference data to the
9 requester.

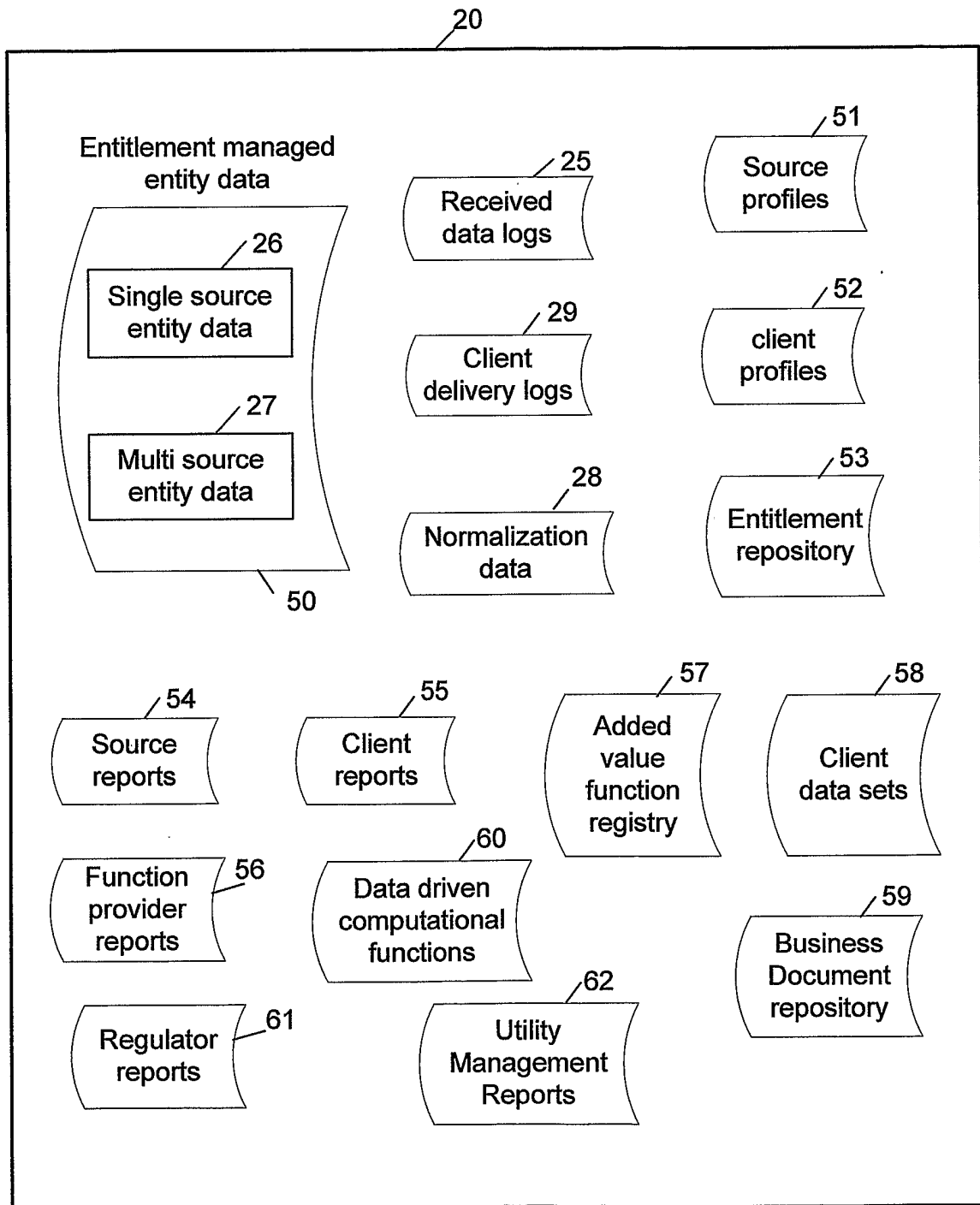
1/47

Fig. 1A



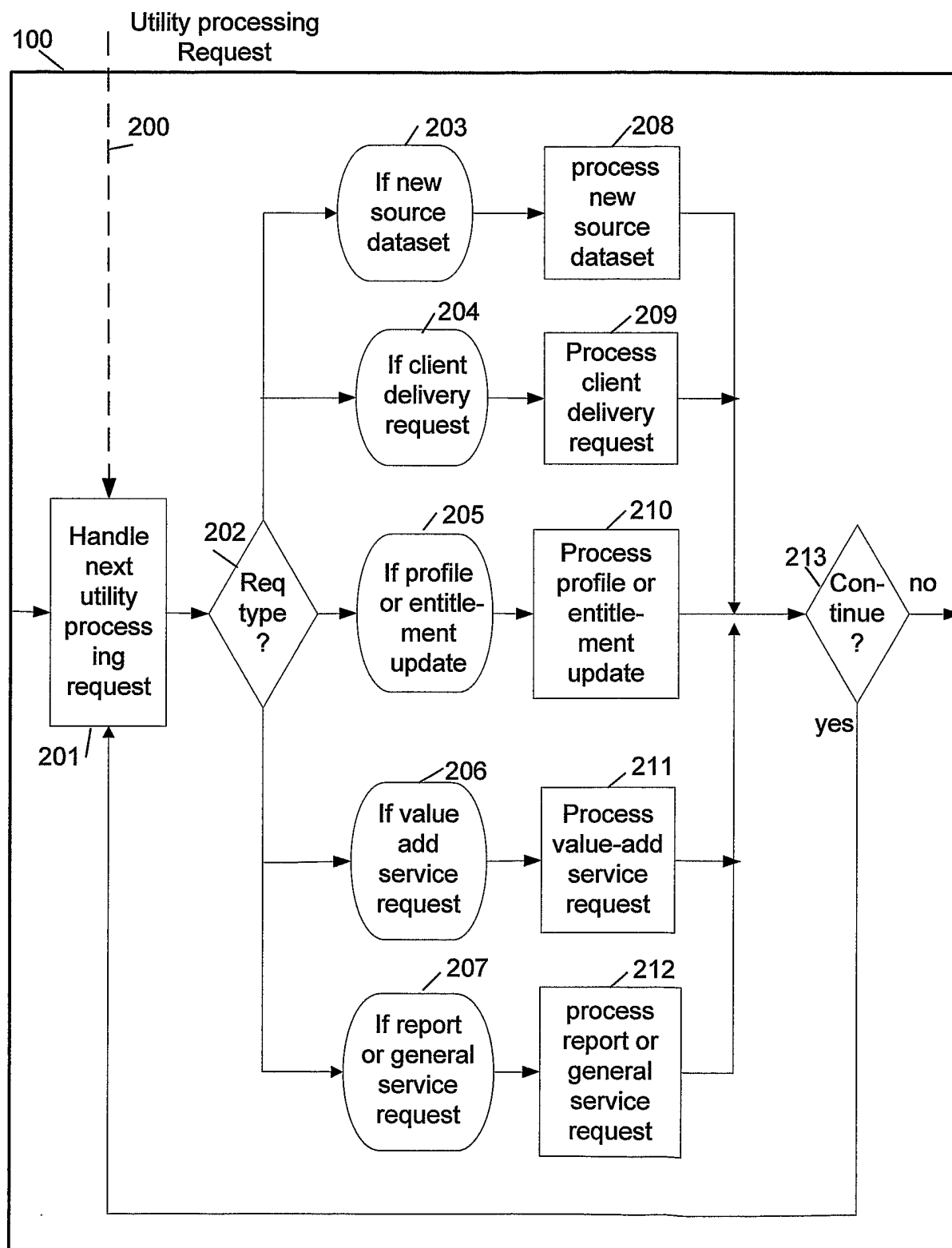
2/47

Fig. 1B



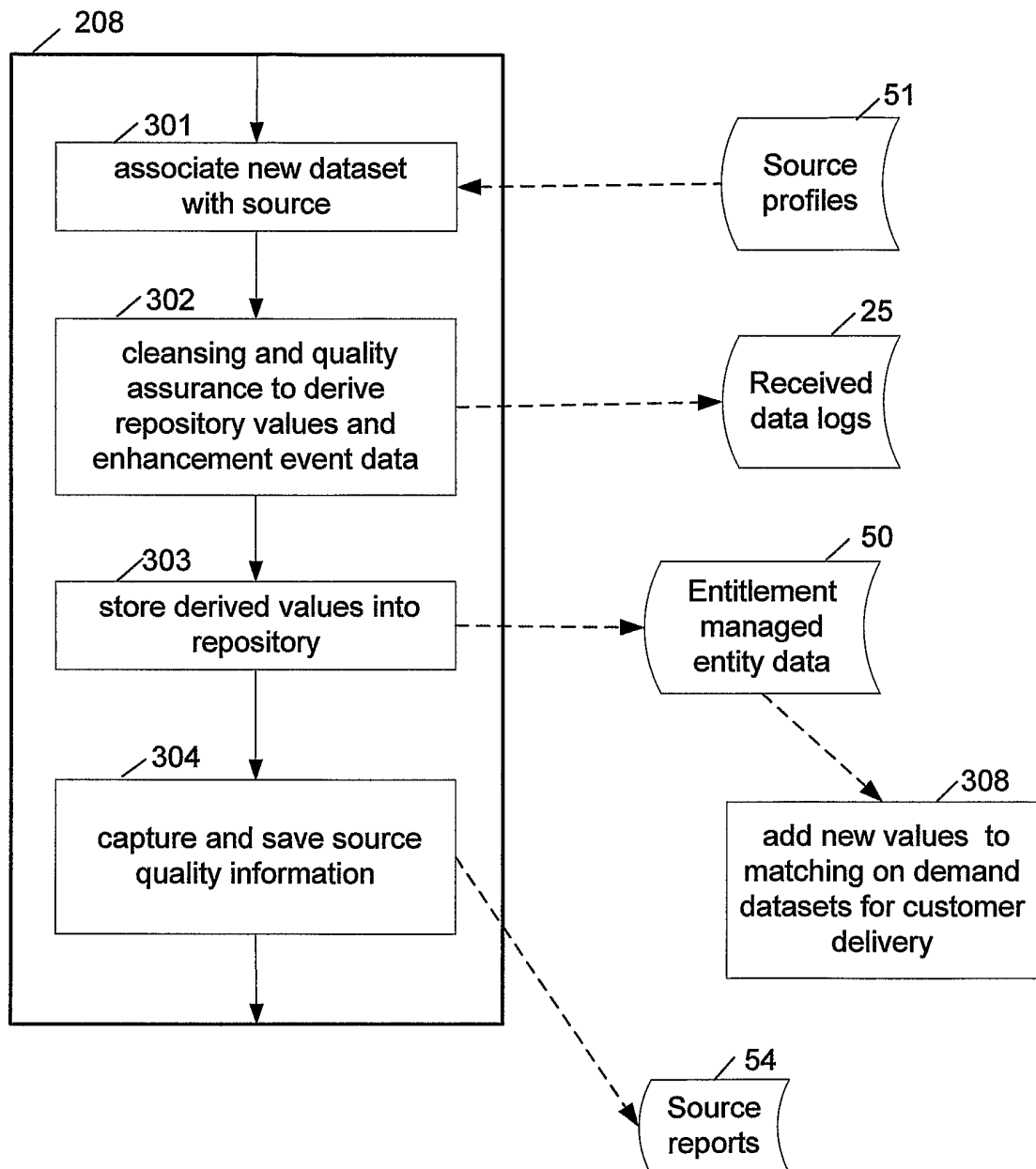
3/47

Fig. 2



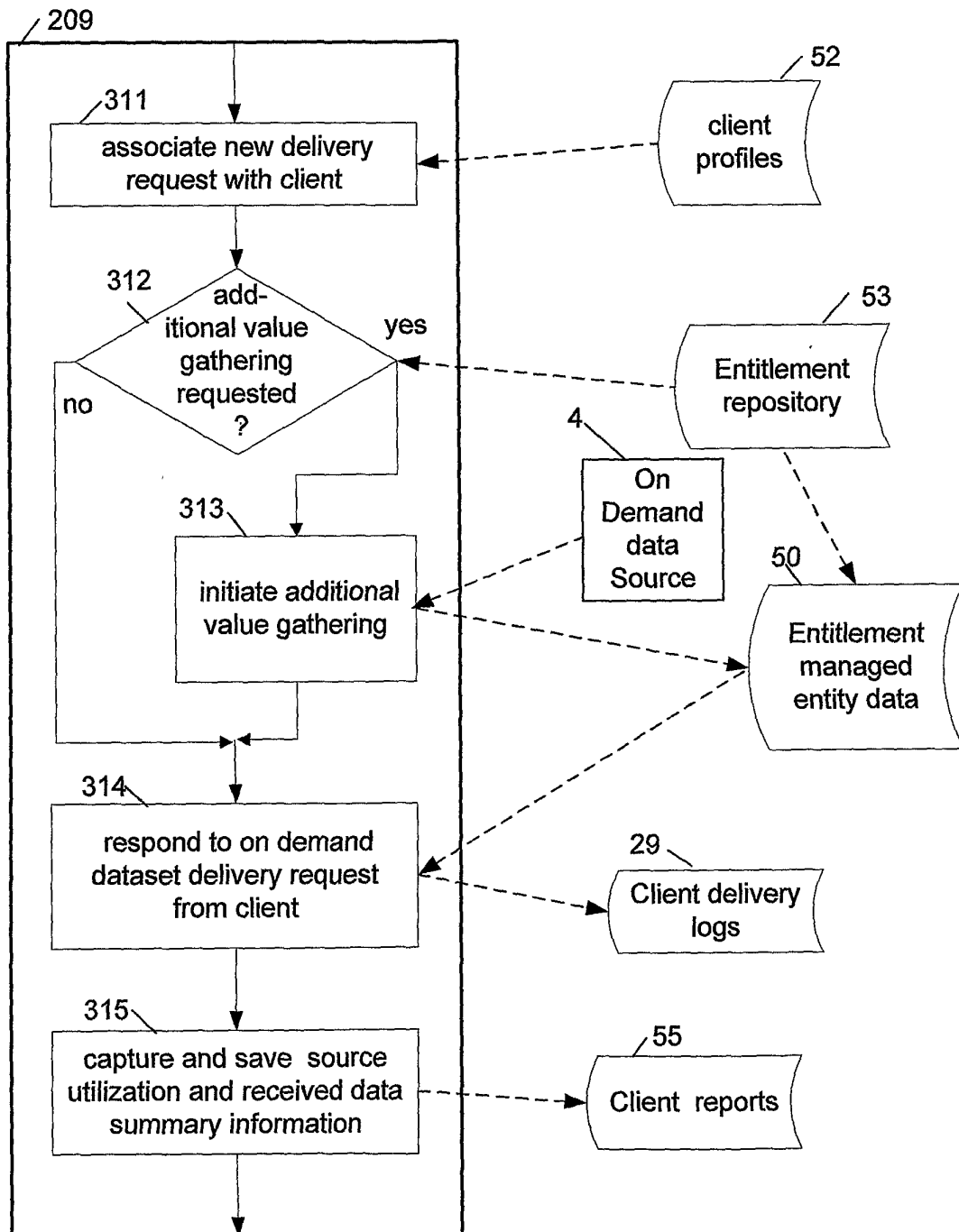
4/47

Fig. 3A



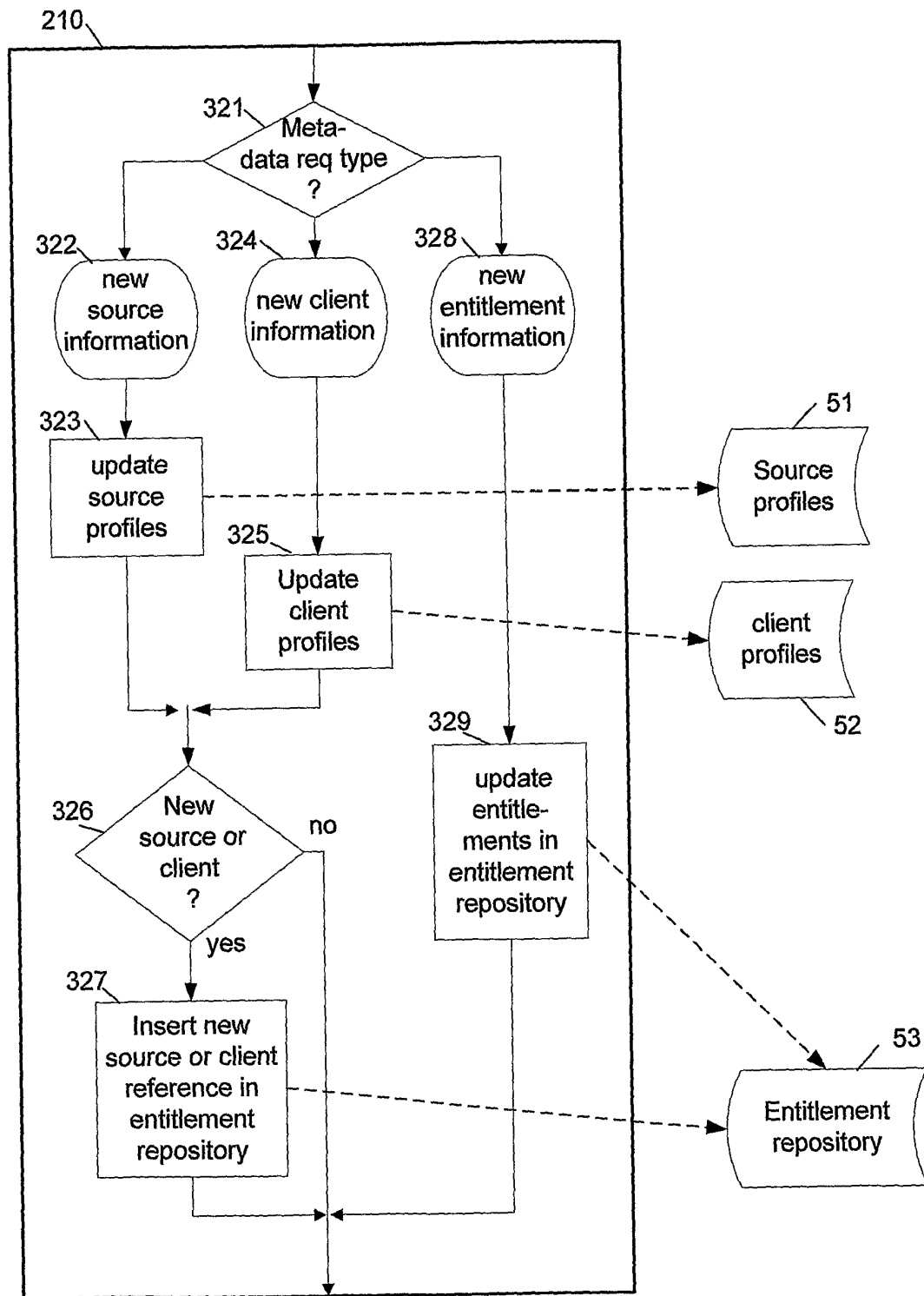
5/47

Fig. 3B



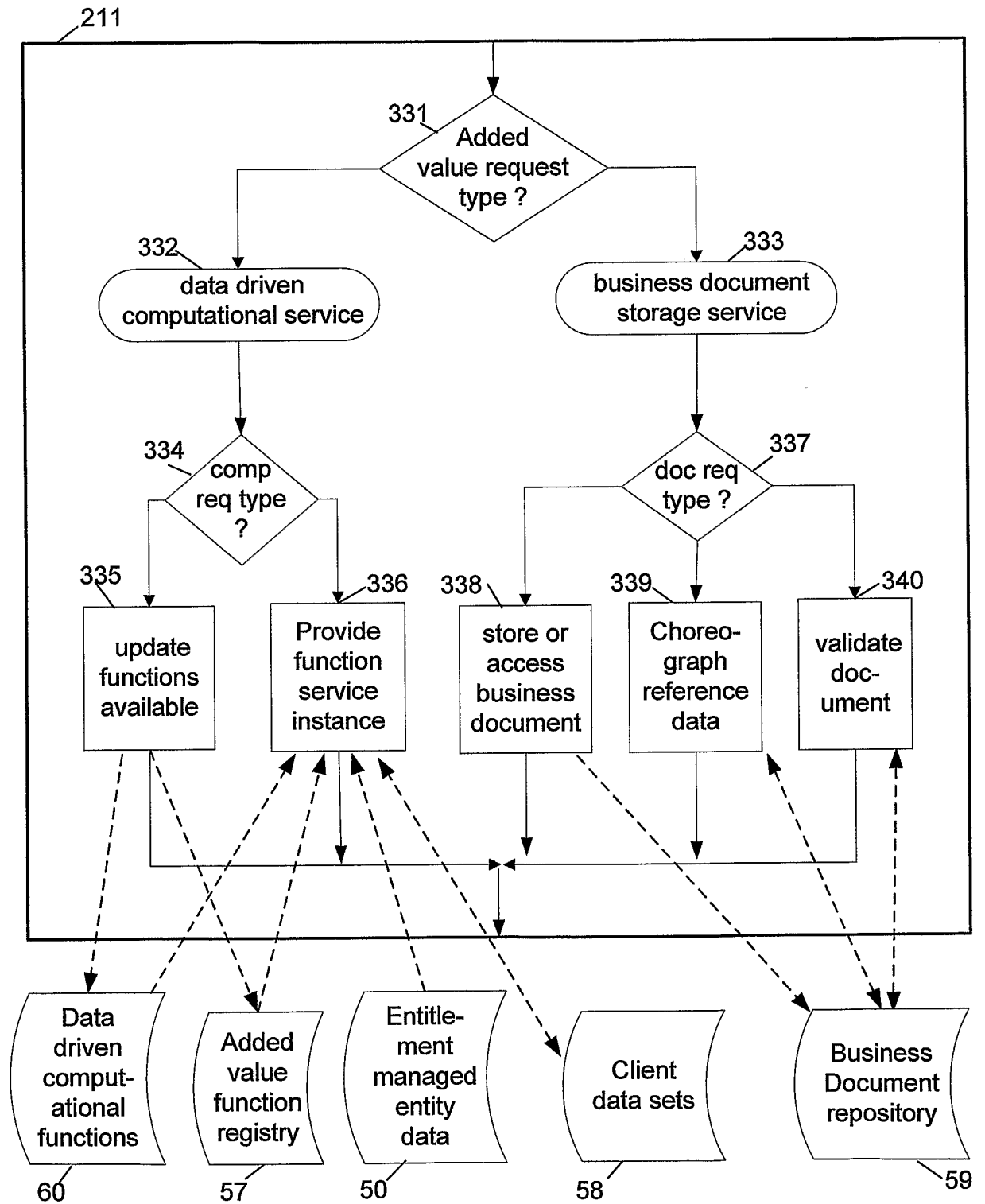
6/47

Fig. 3C



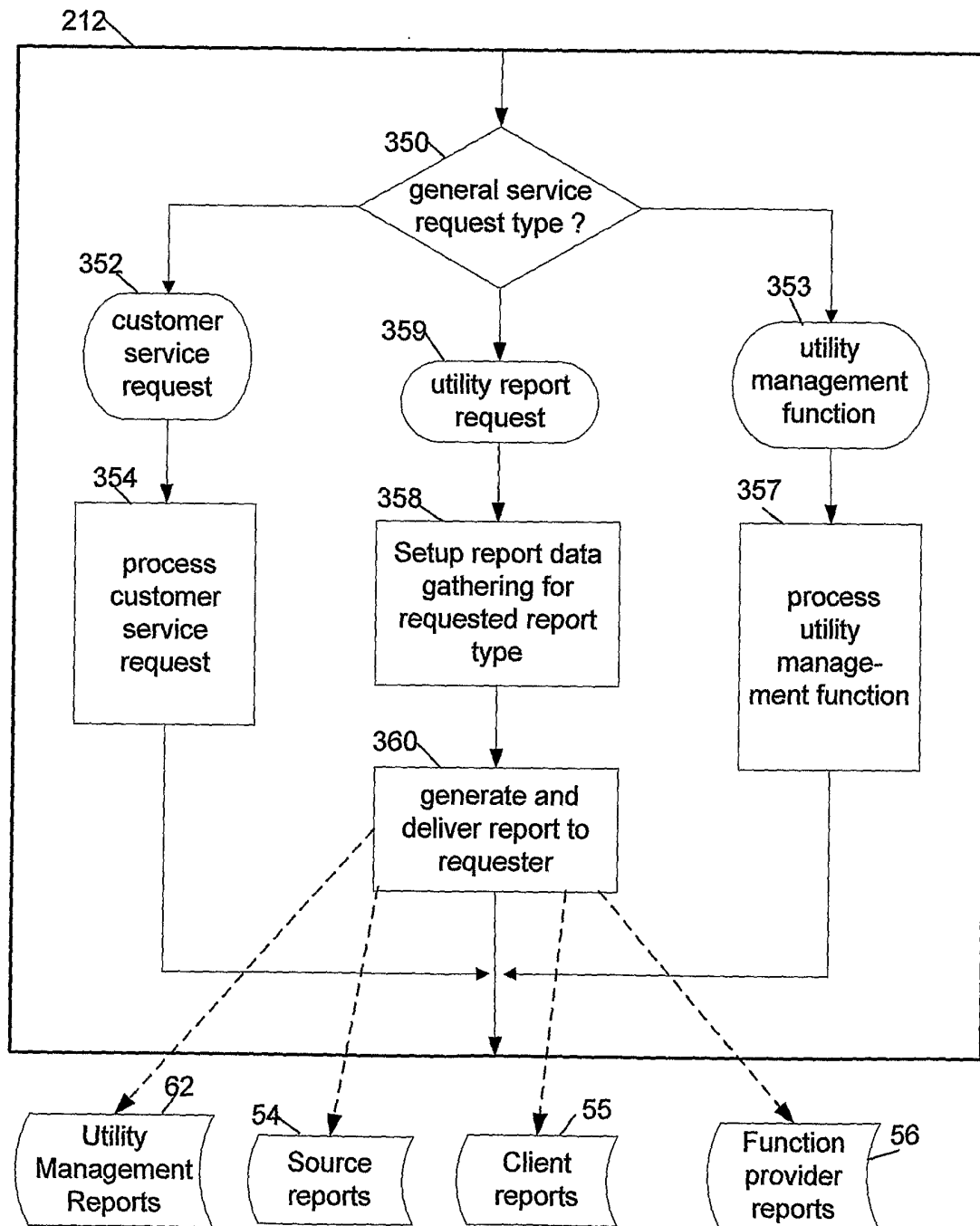
7/47

Fig. 3D



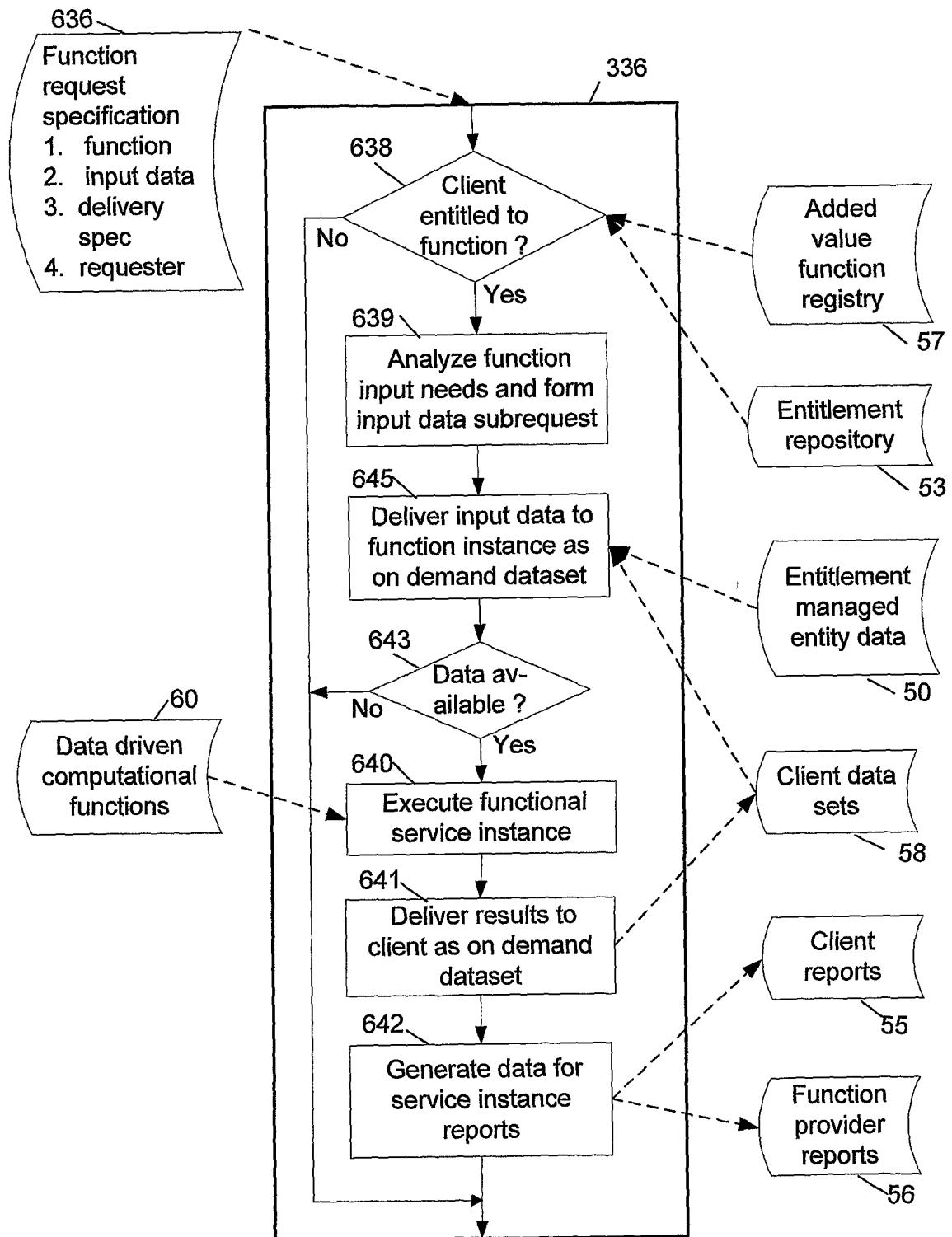
8/47

Fig. 3E



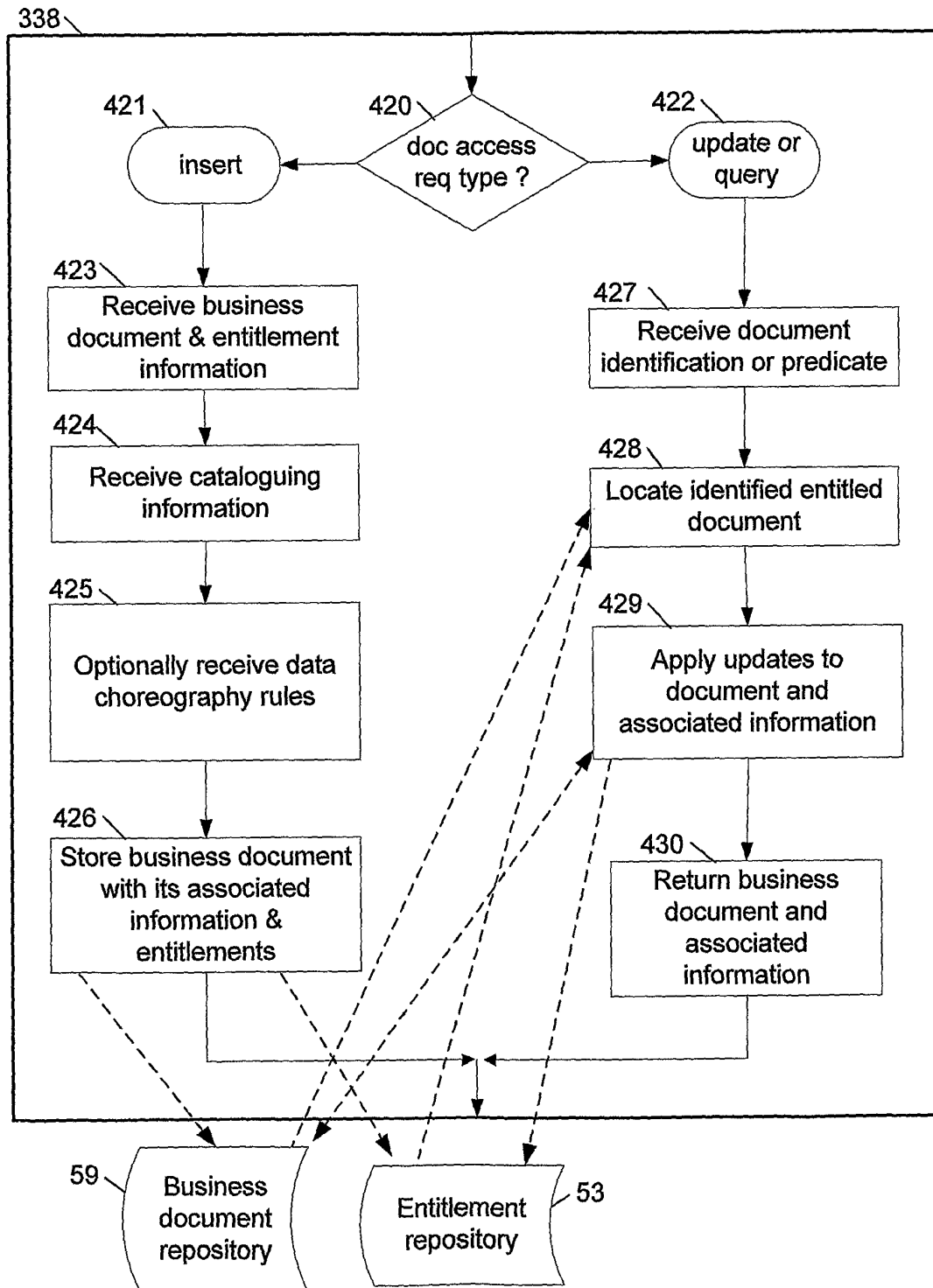
9/47

Fig 4A



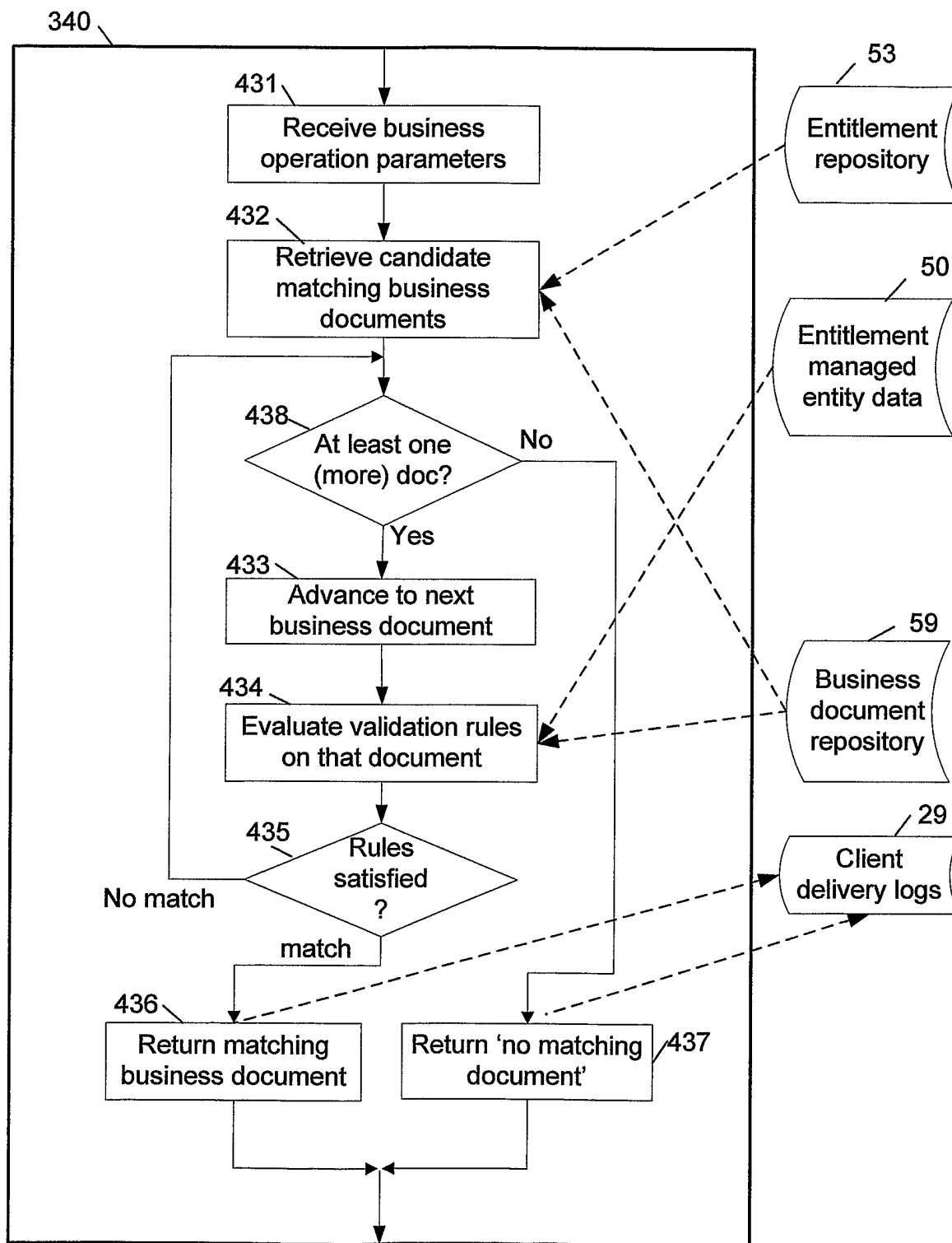
10/47

Fig. 4B



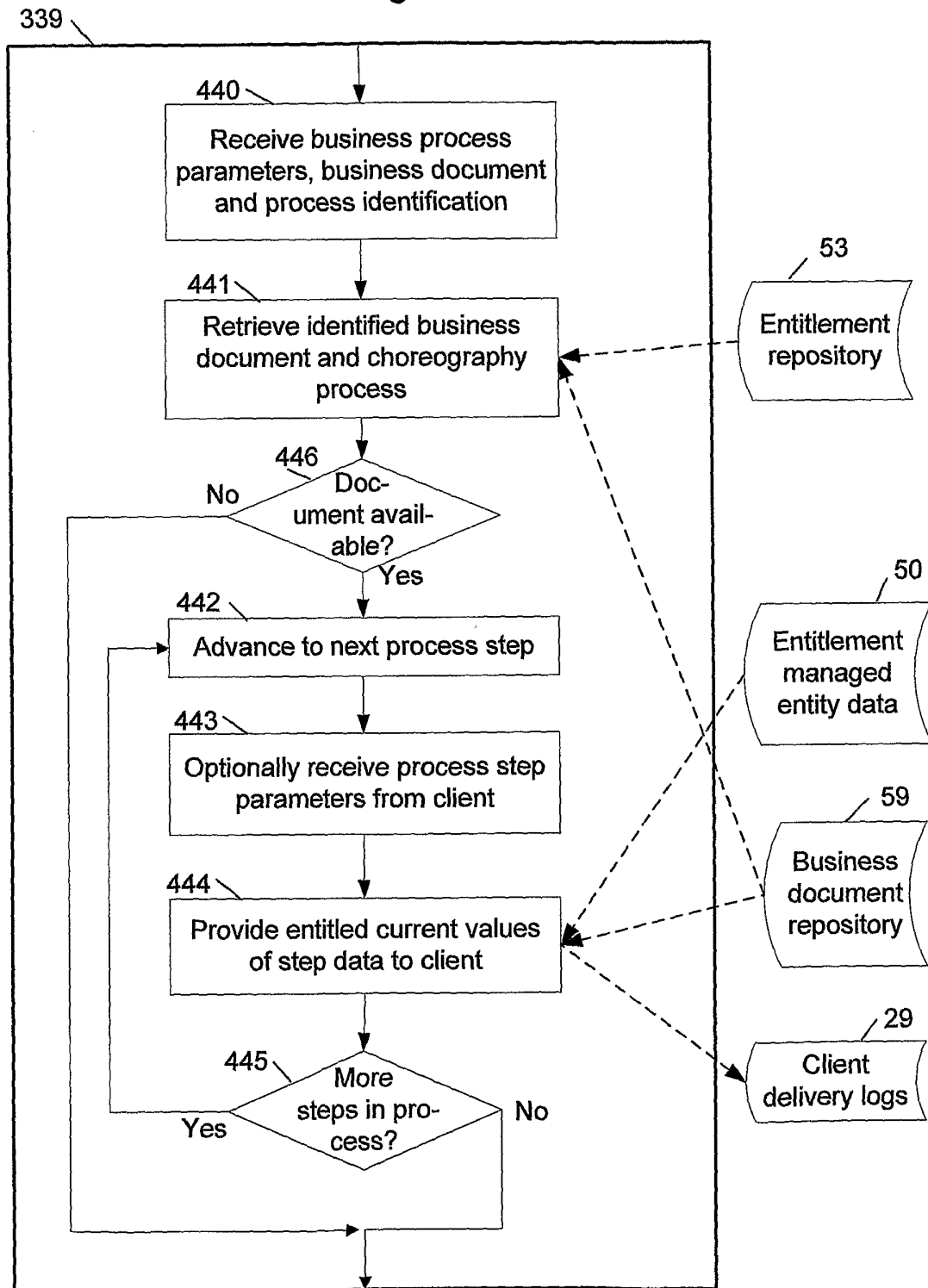
11/47

Fig. 4C



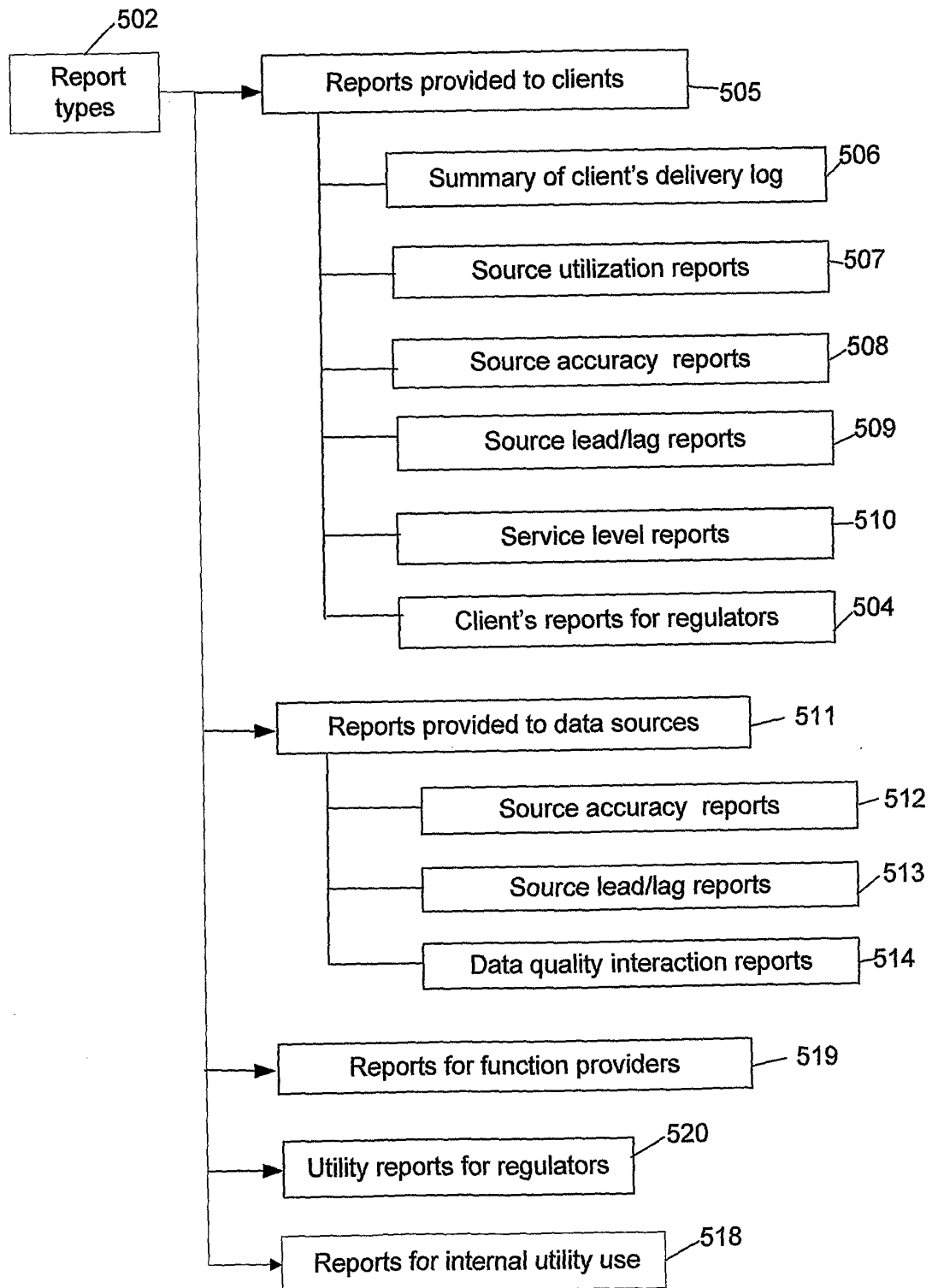
12/47

Fig. 4D



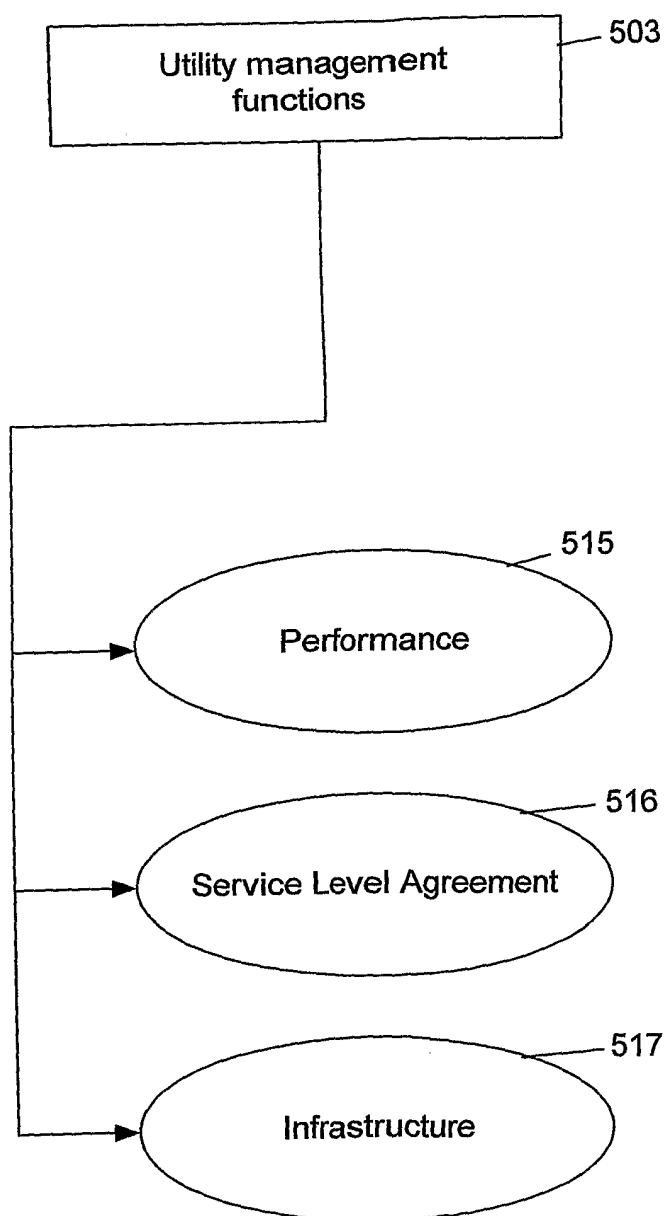
13/47

Fig. 5A



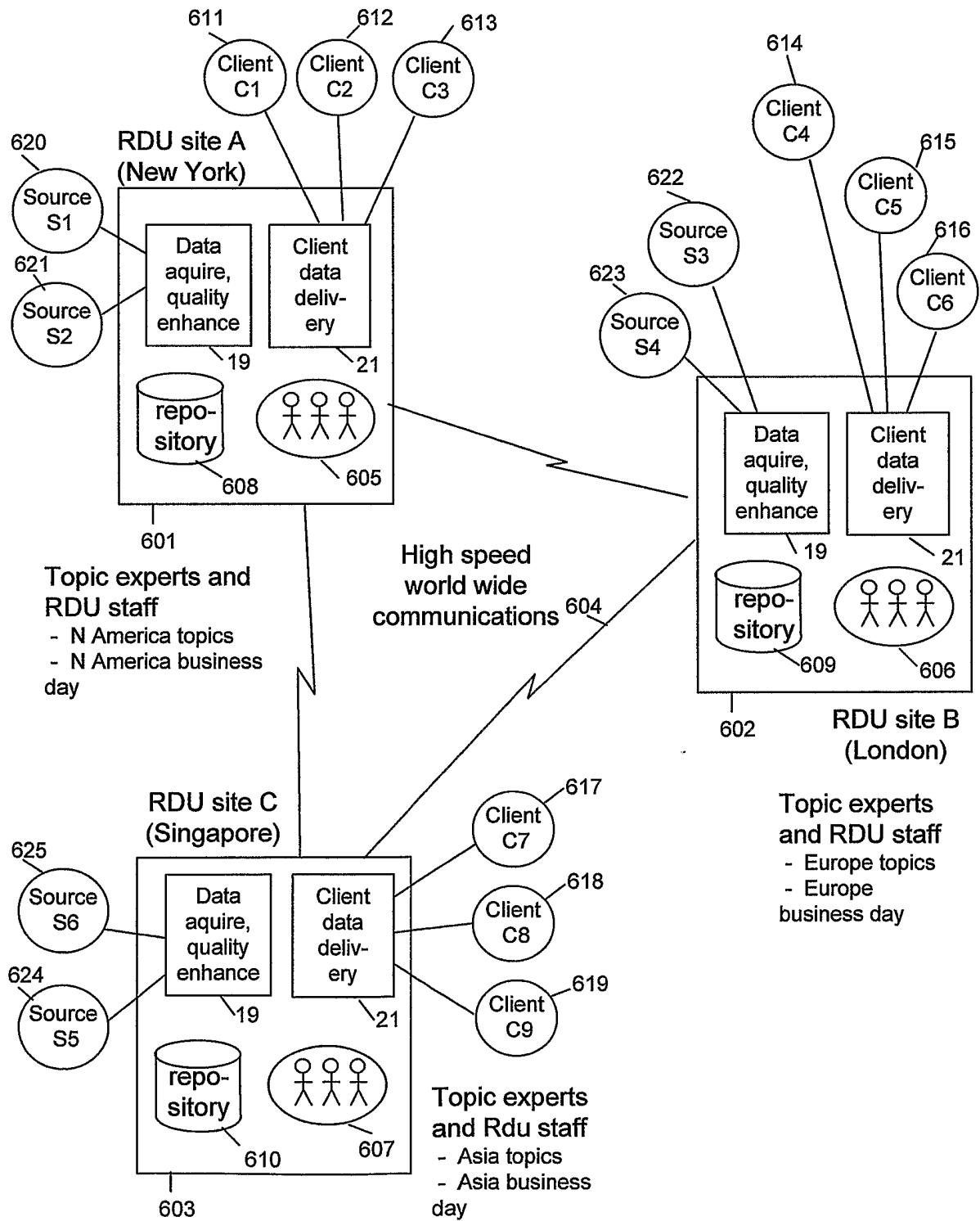
14/47

Fig. 5B



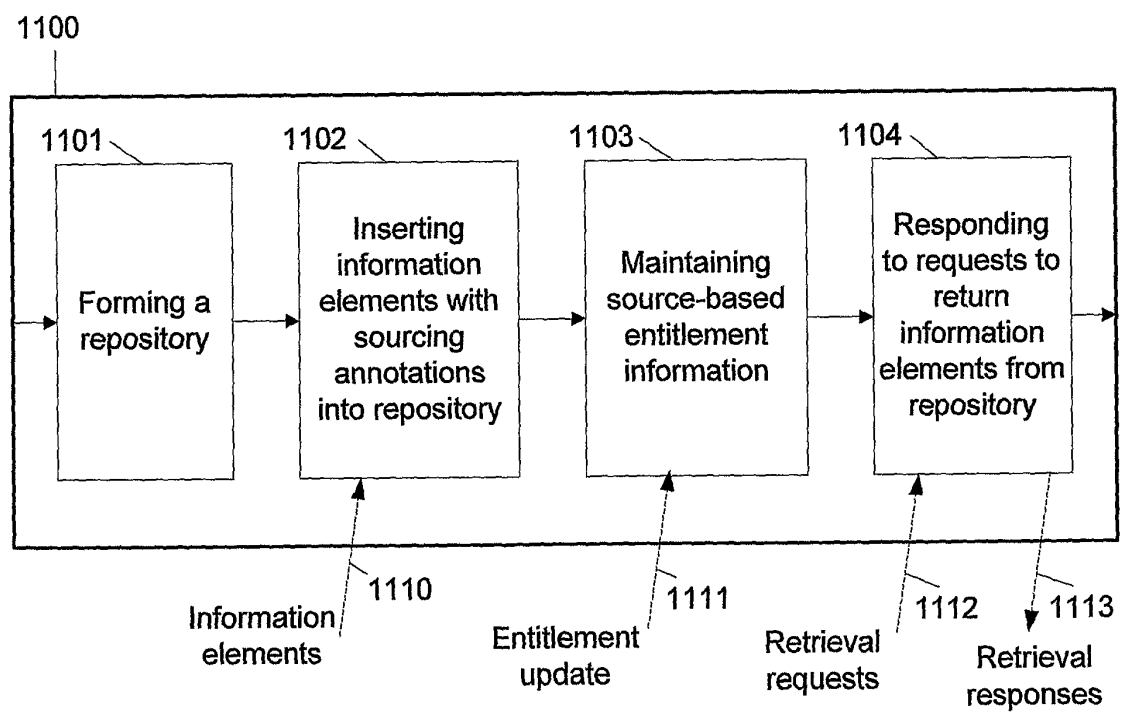
15/47

Fig. 6



16/47

Fig. 7A



17/47

Fig. 7B

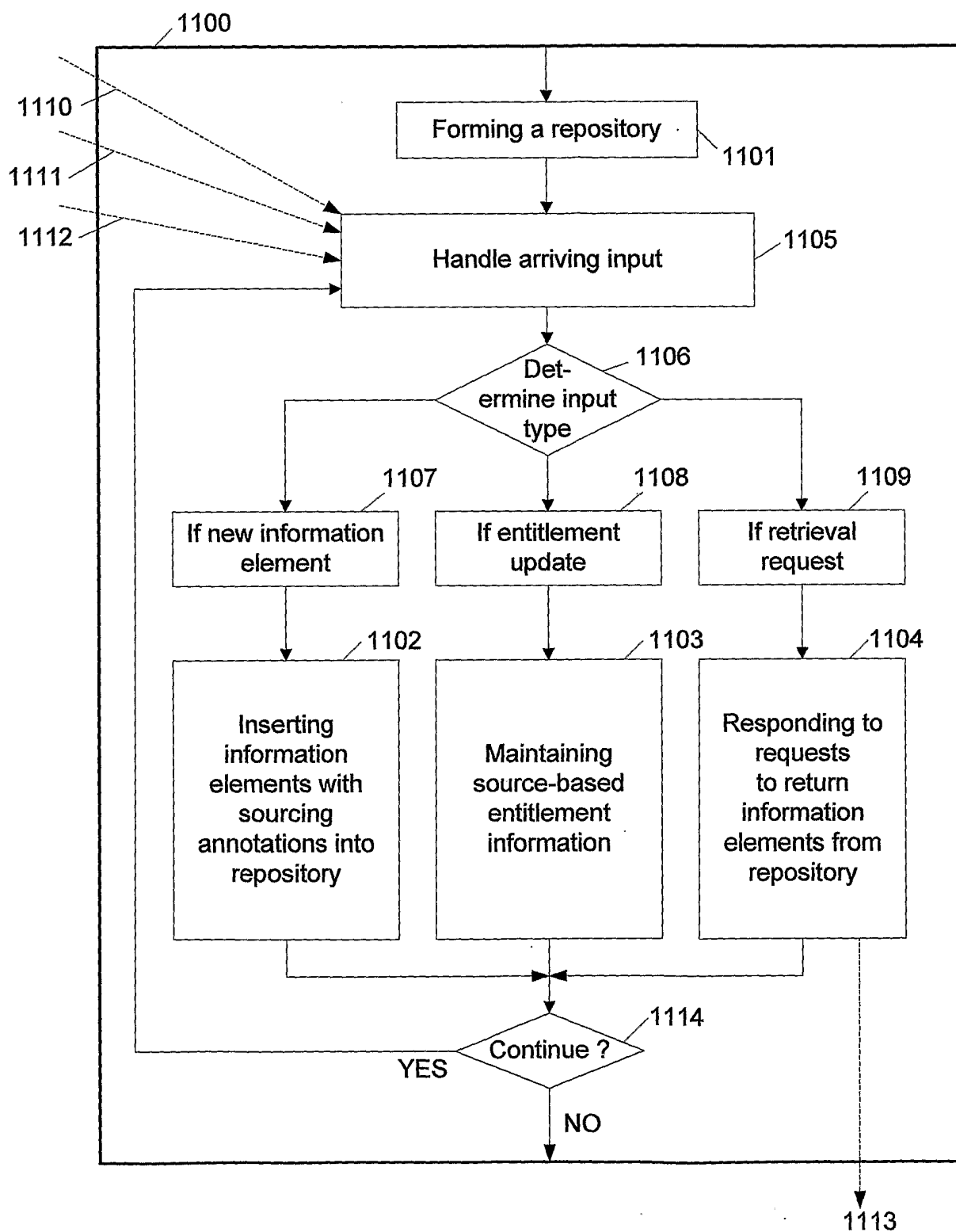


Fig. 8A

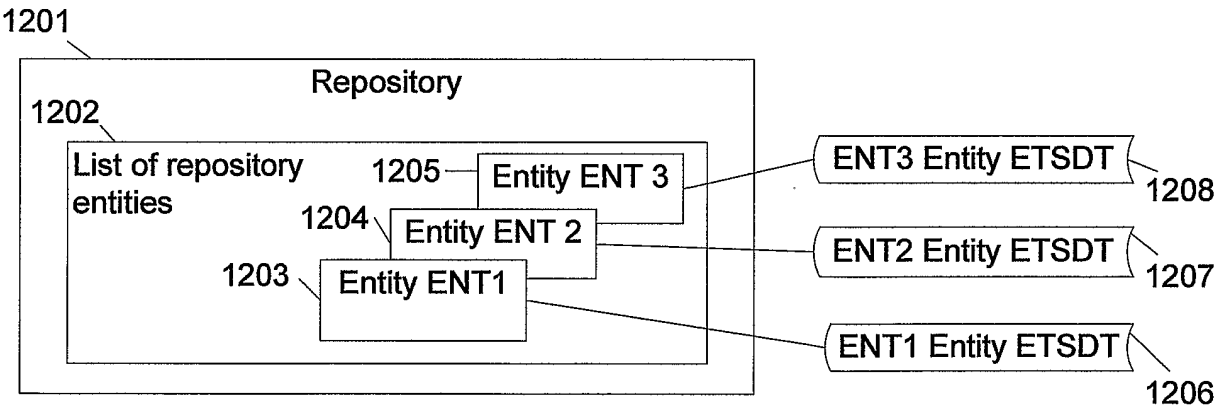
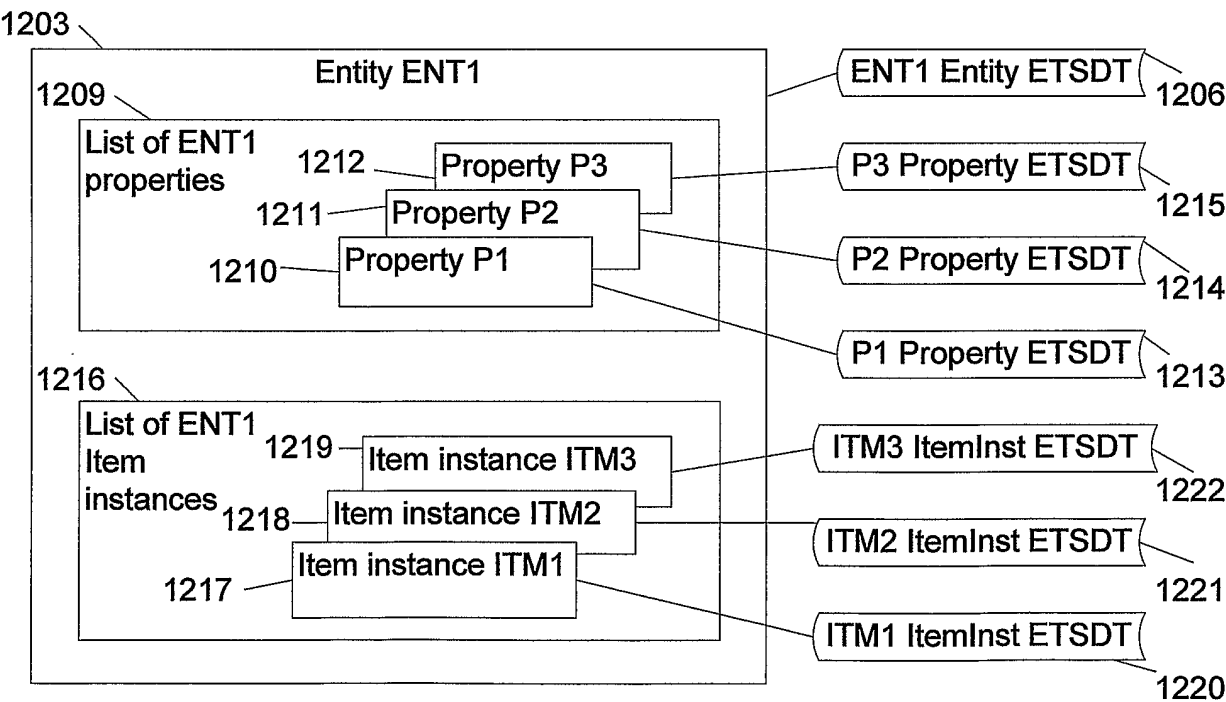


Fig. 8B



20/47

Fig. 8C

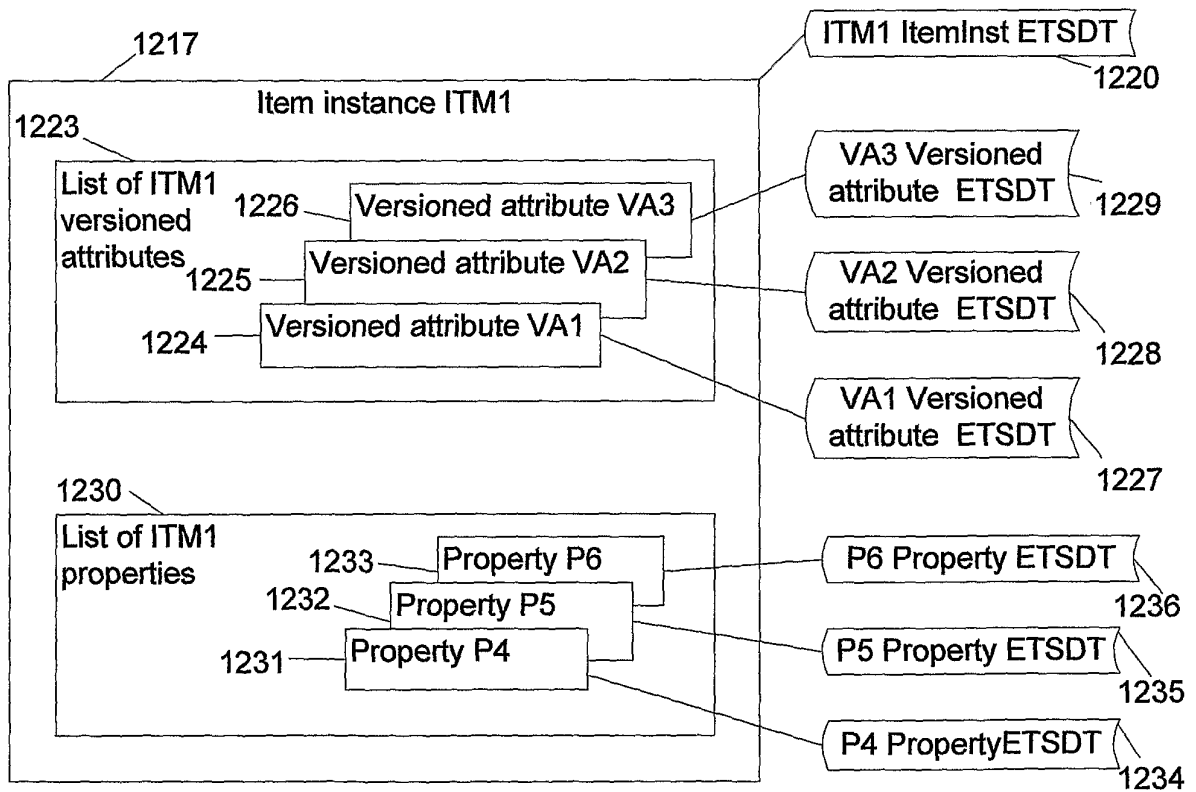
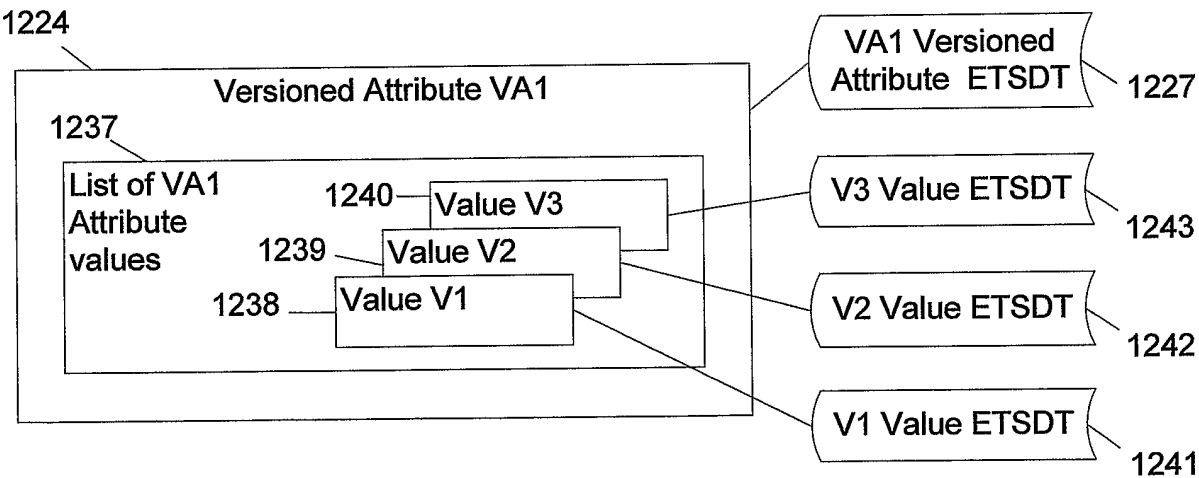
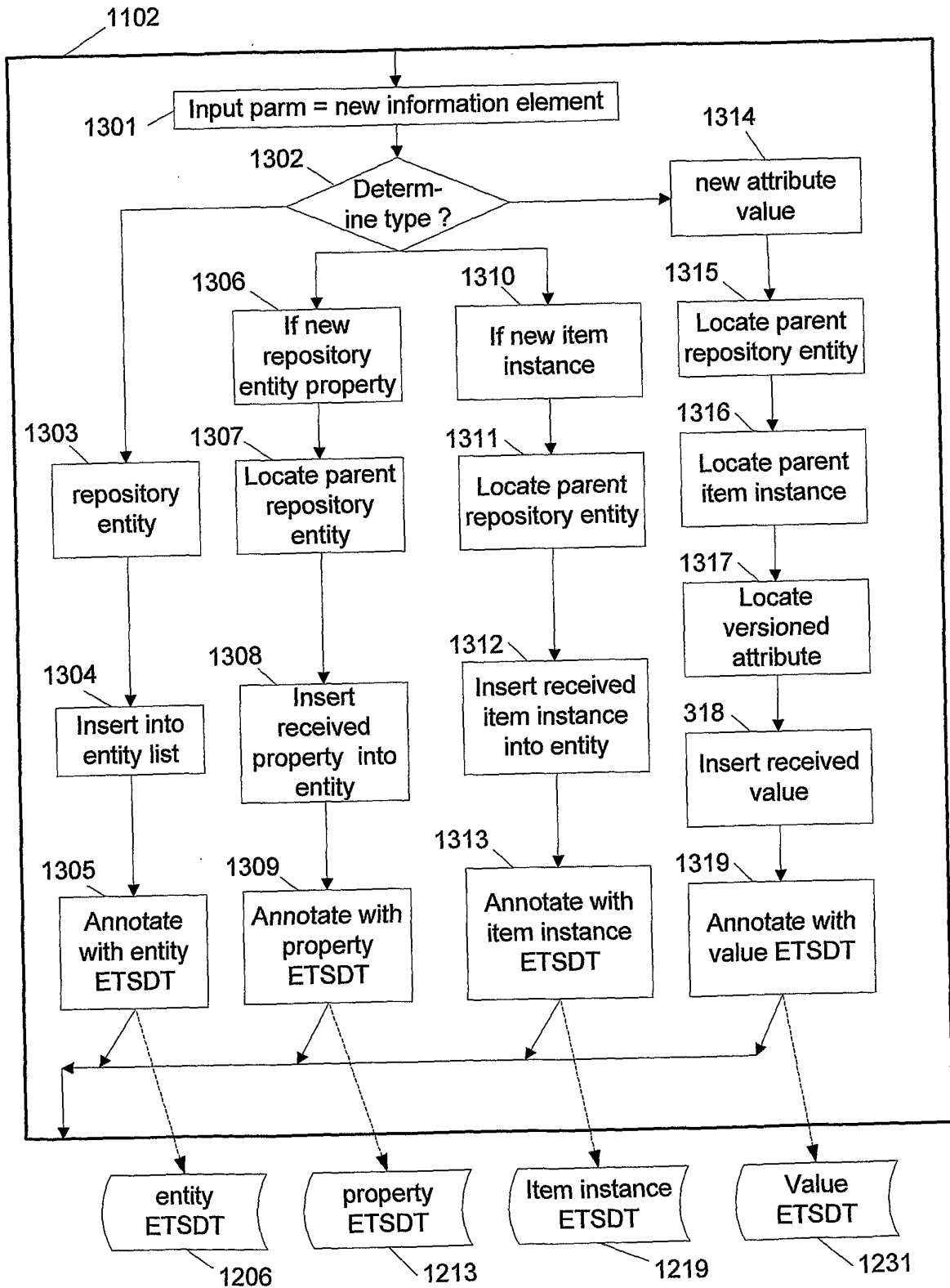


Fig. 8D



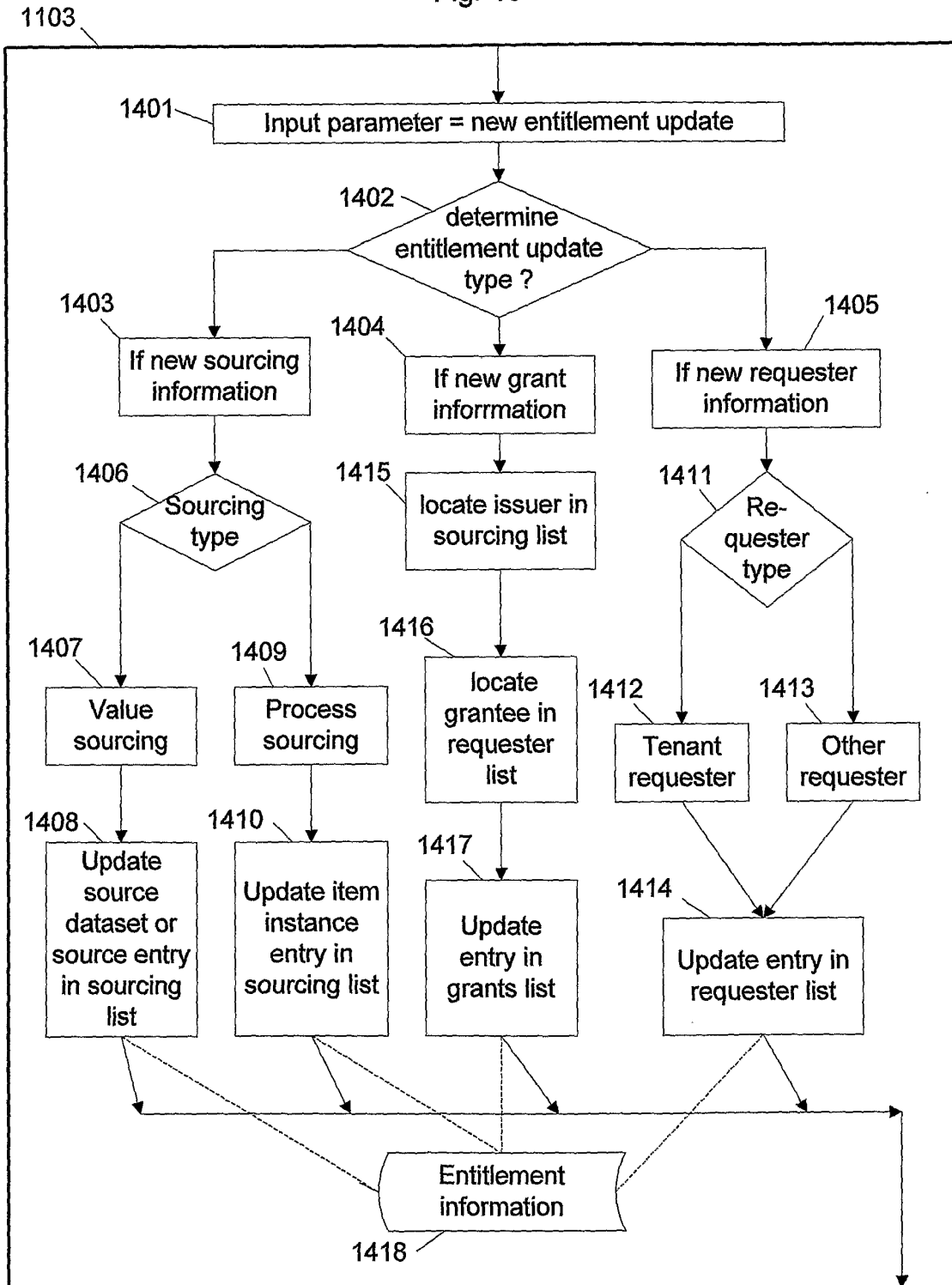
22/47

Fig. 9



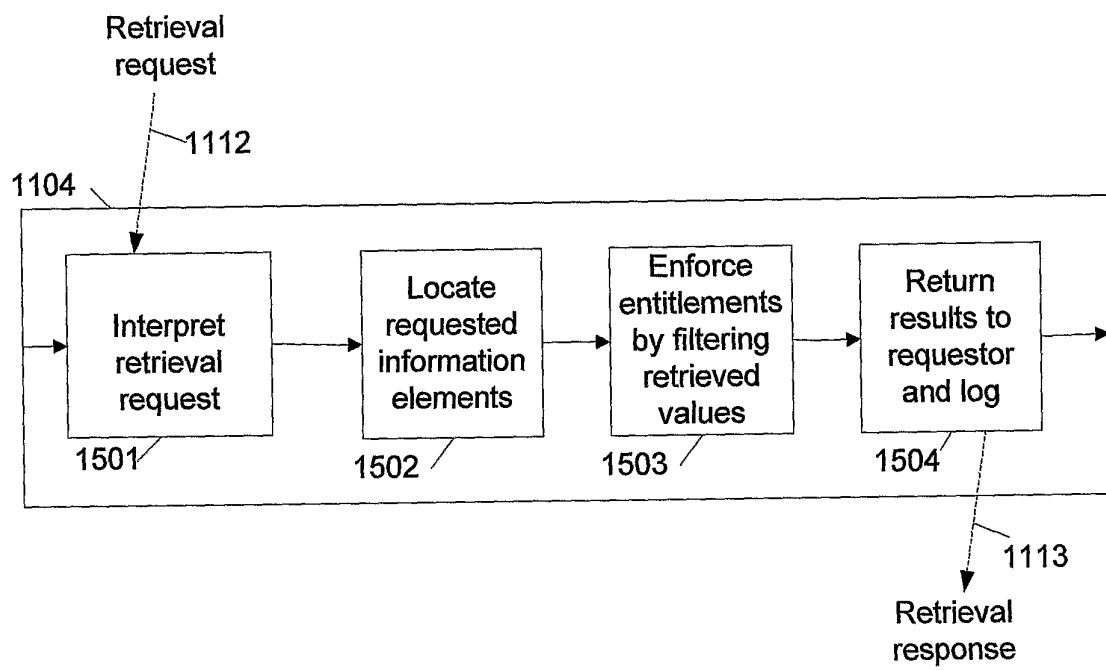
23/47

Fig. 10



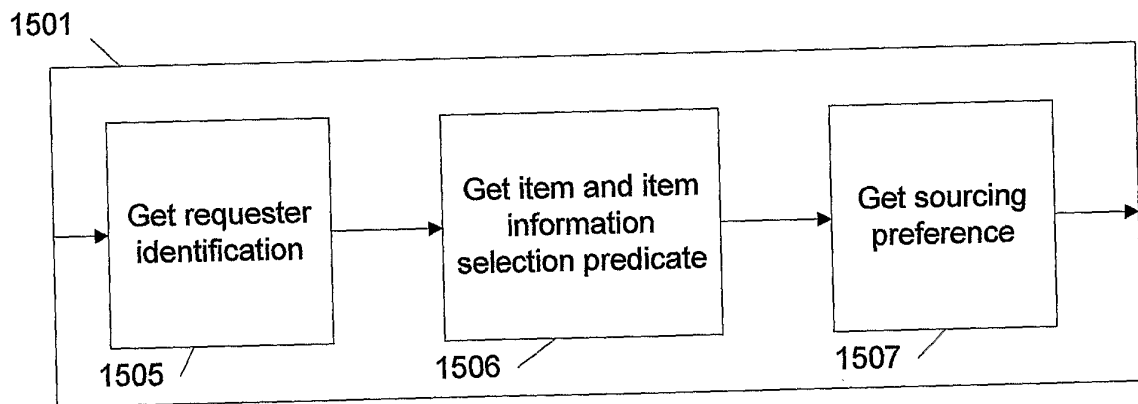
24/47

Fig. 11A



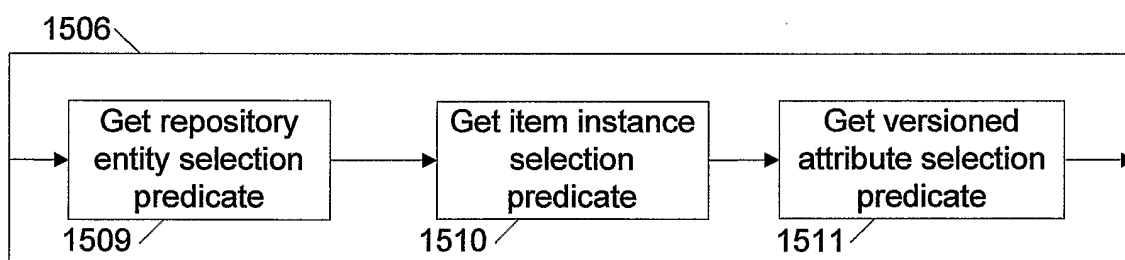
25/47

Fig. 11B



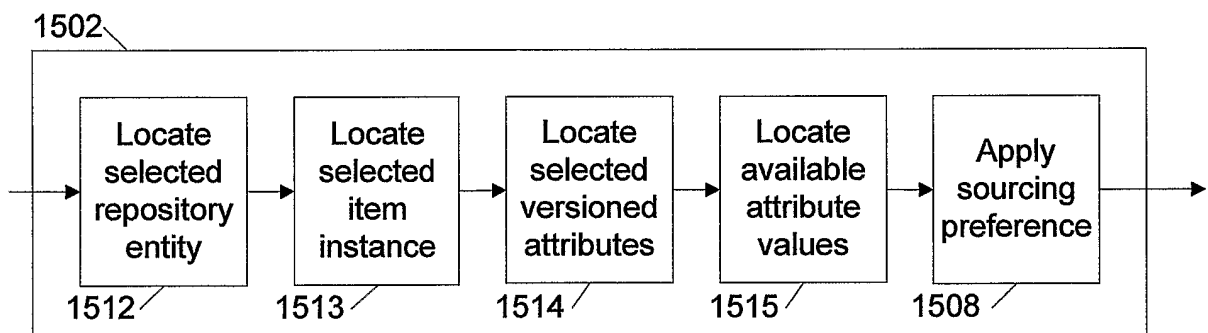
26/47

Fig. 11C



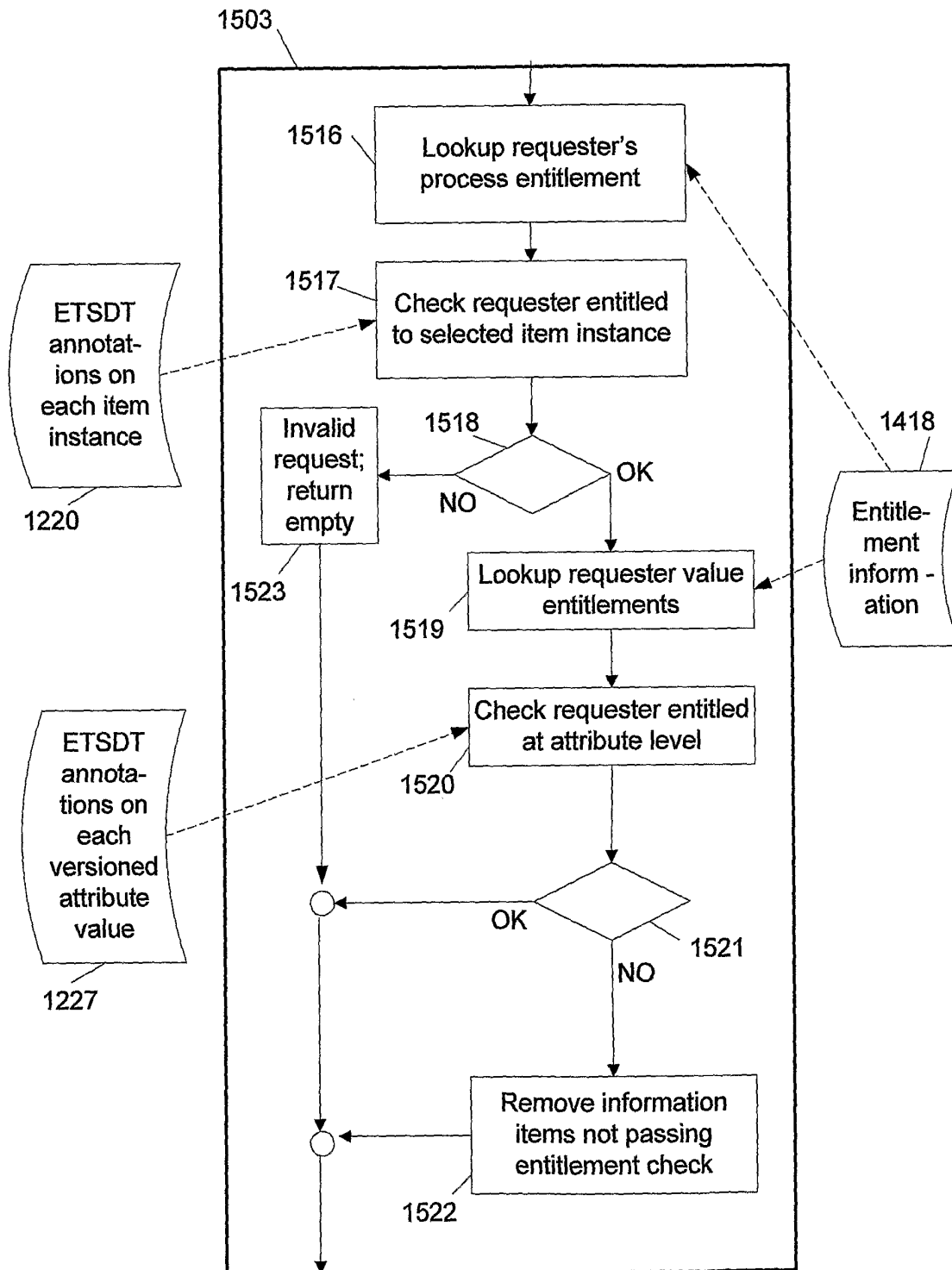
27/47

Fig. 11D



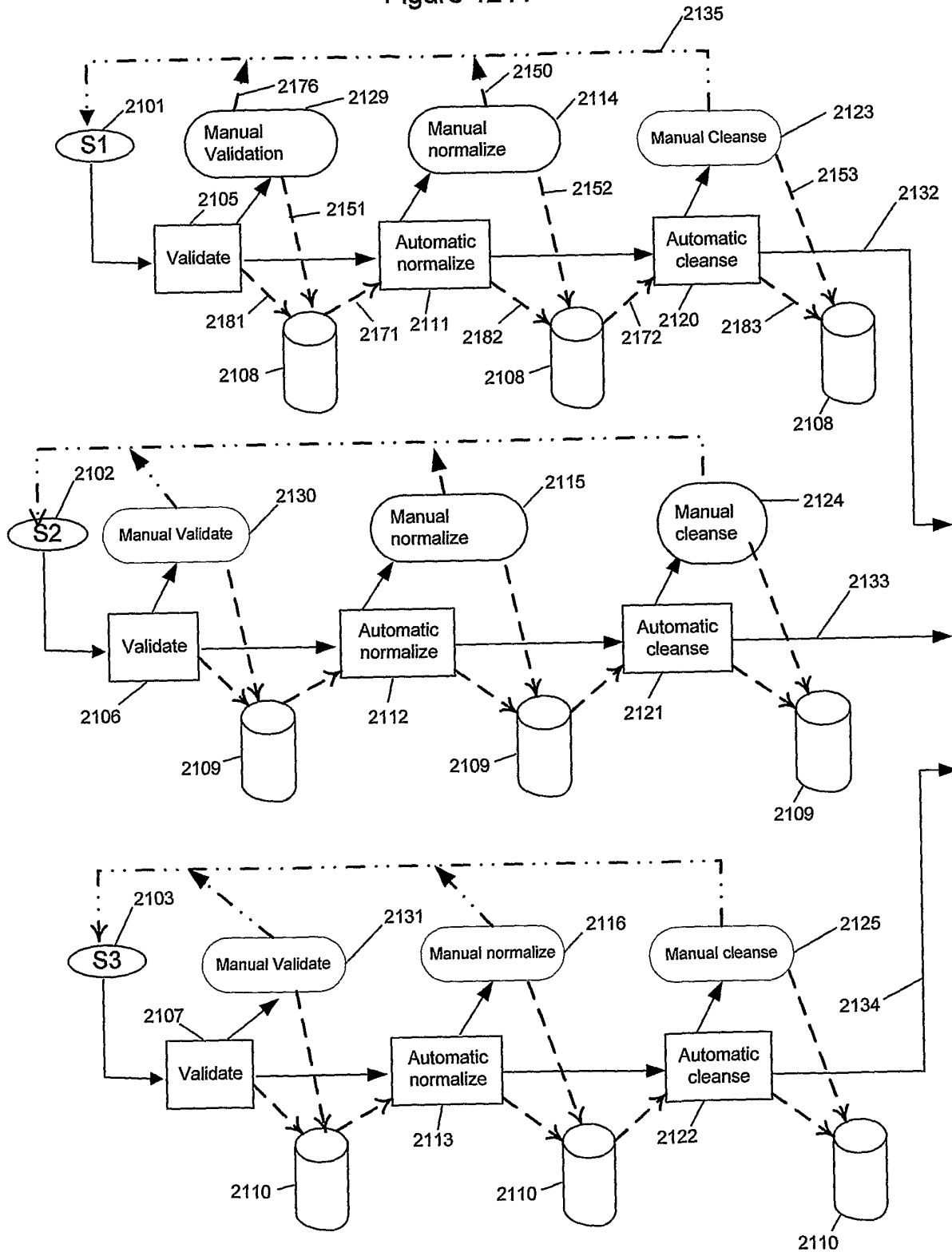
28/47

Fig. 11E



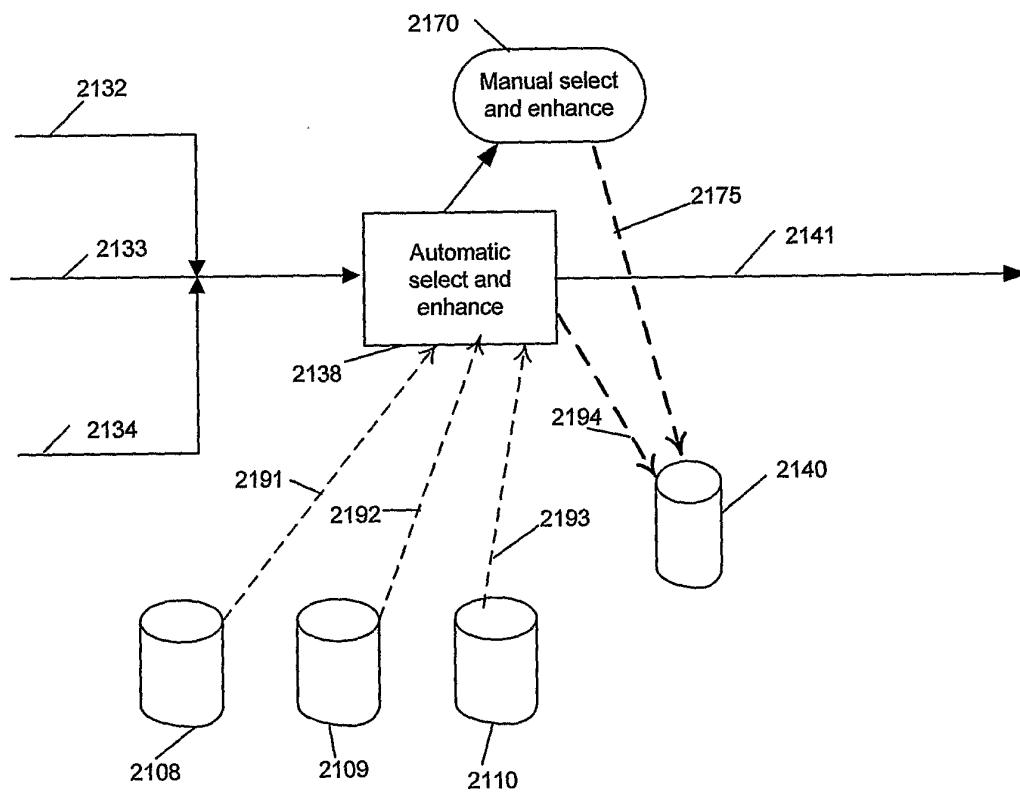
29/47

Figure 12 A



30/47

Figure 12 B



31/47

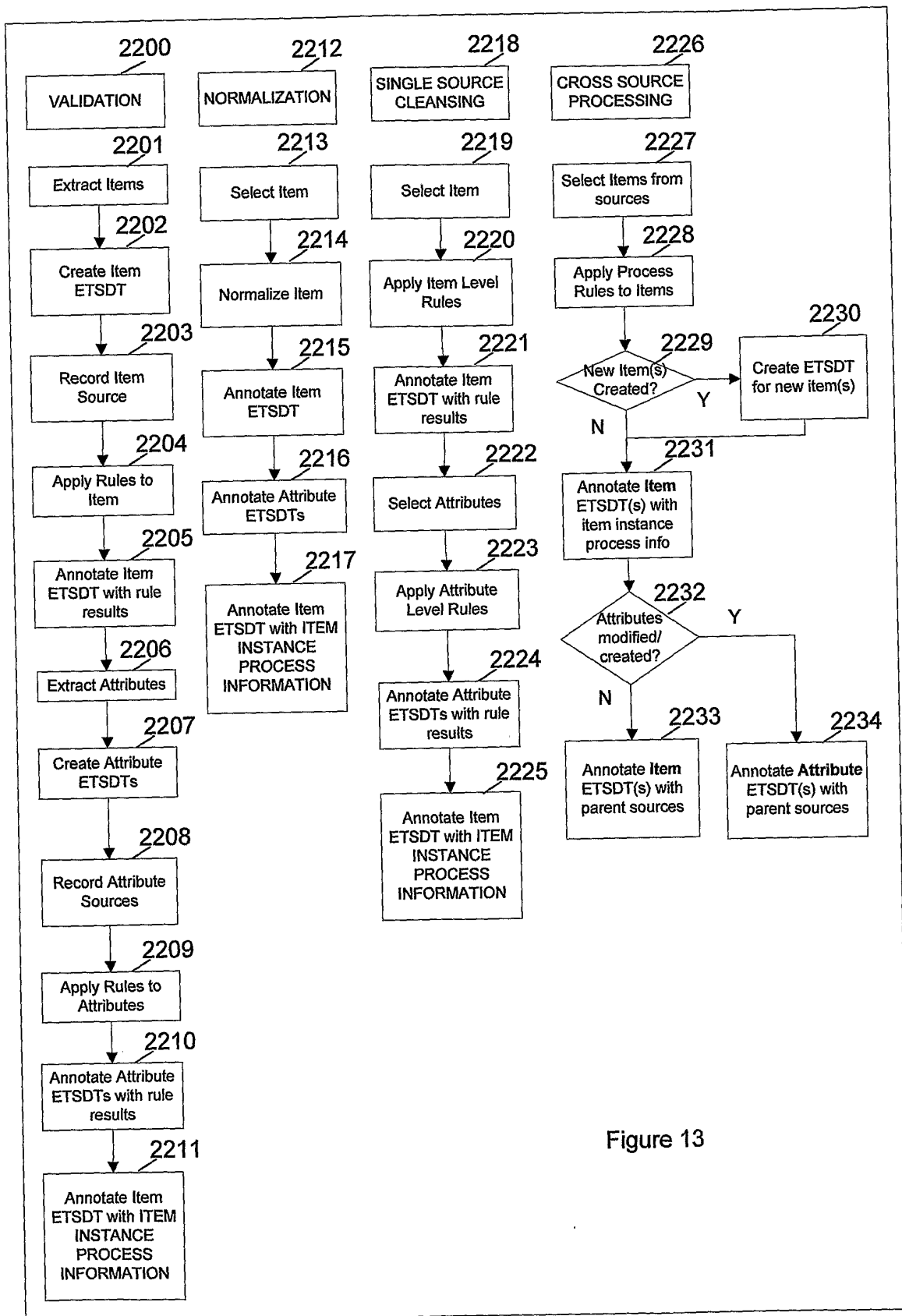
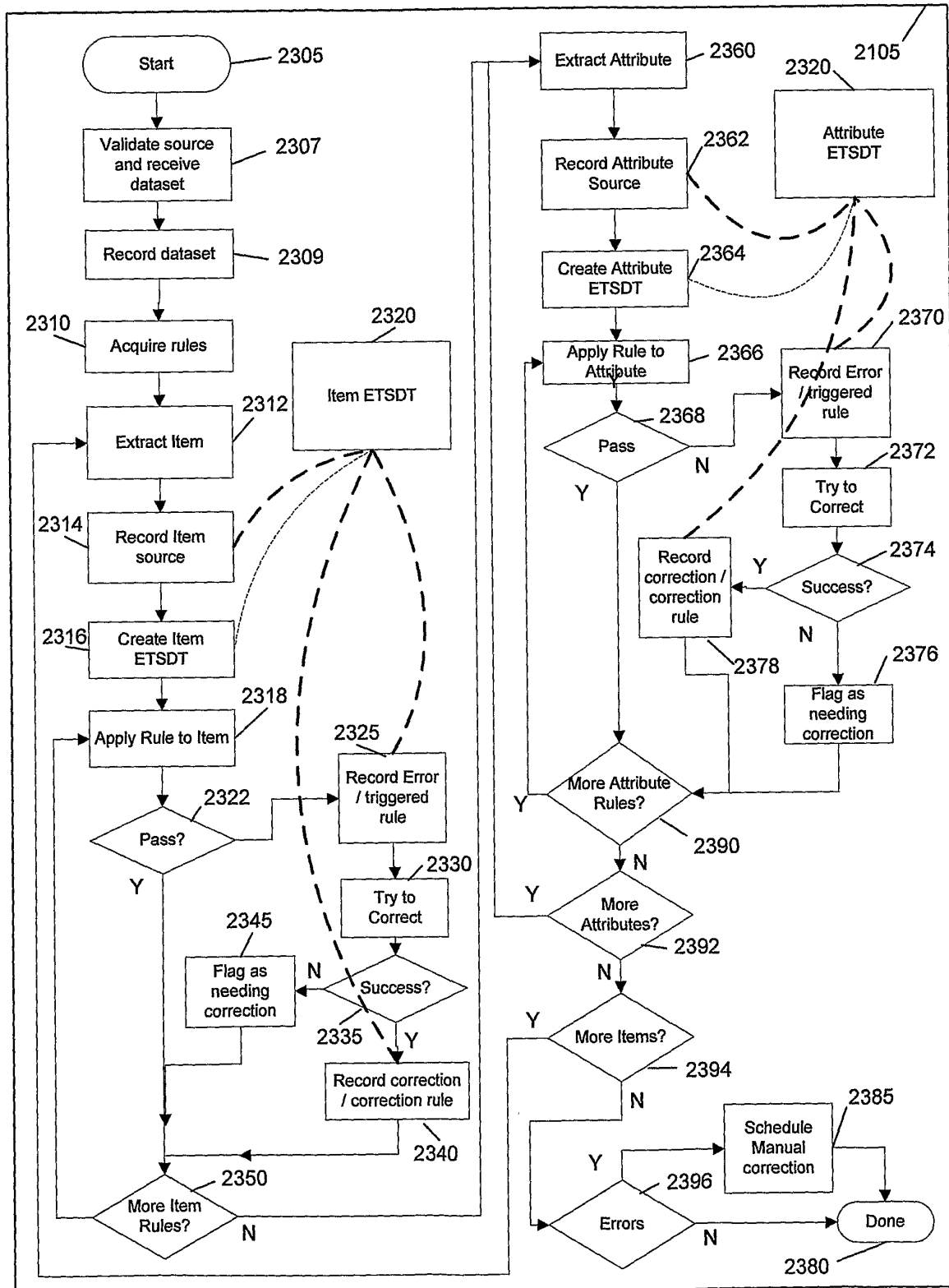


Figure 13

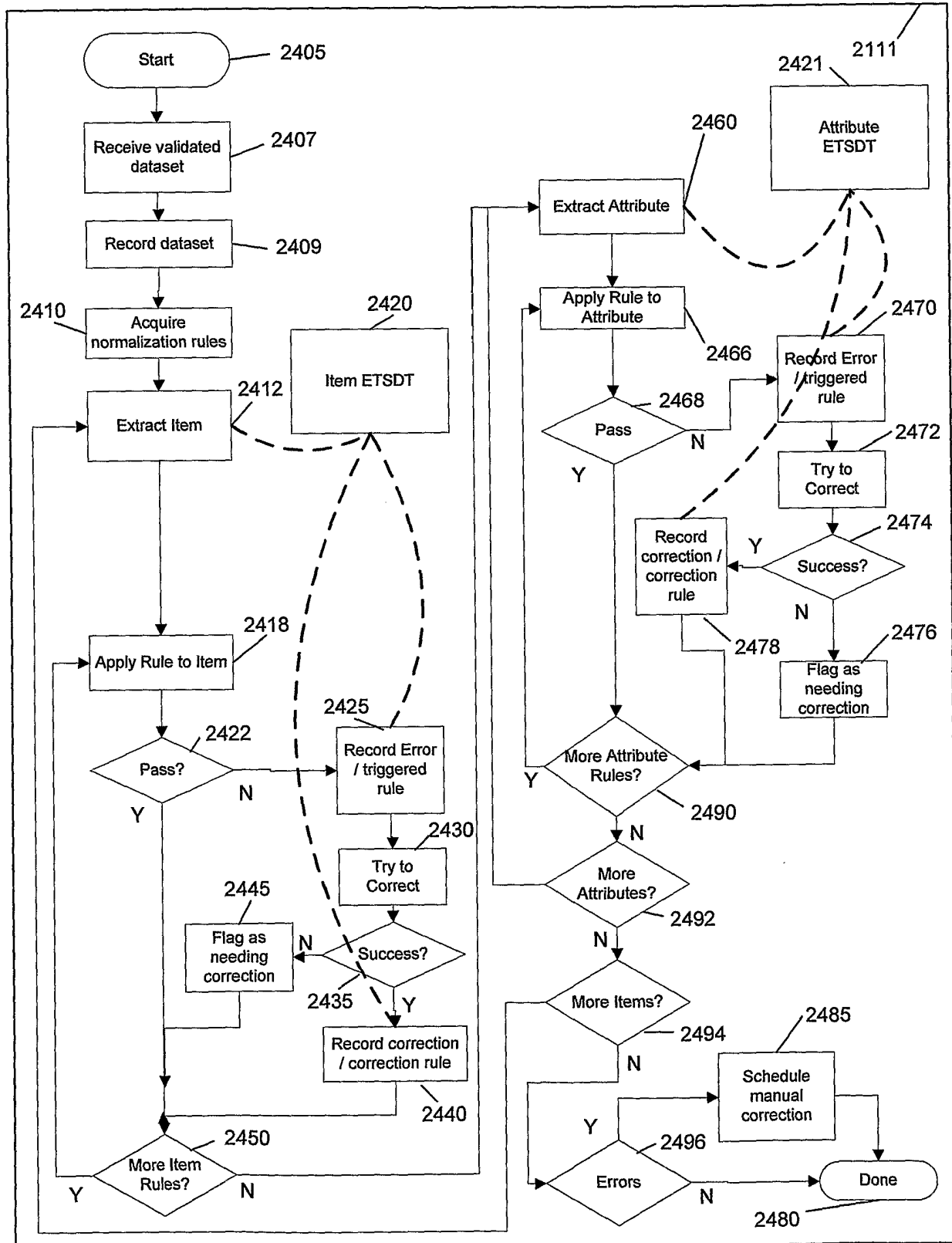
32/47

Figure 14



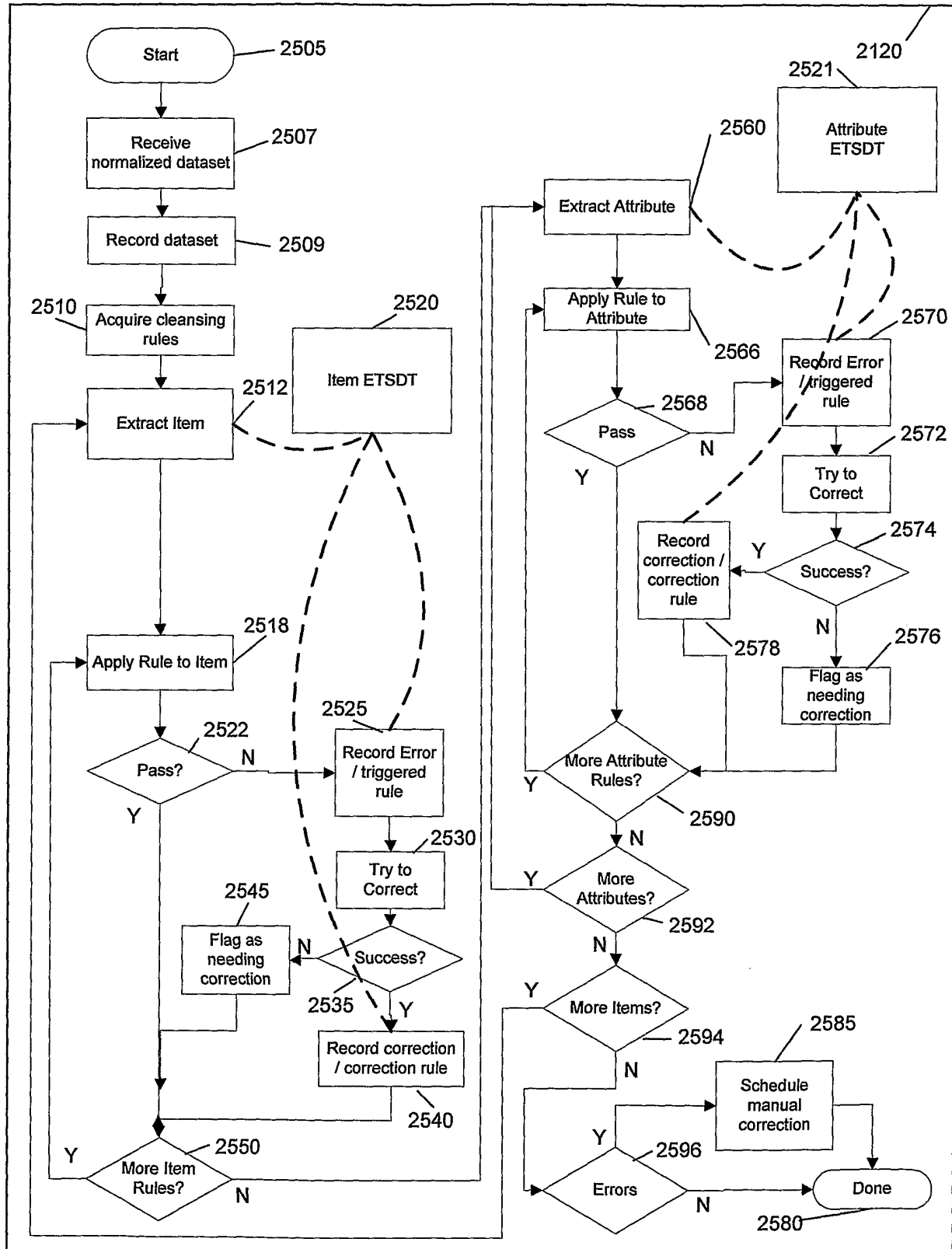
33/47

Figure 15



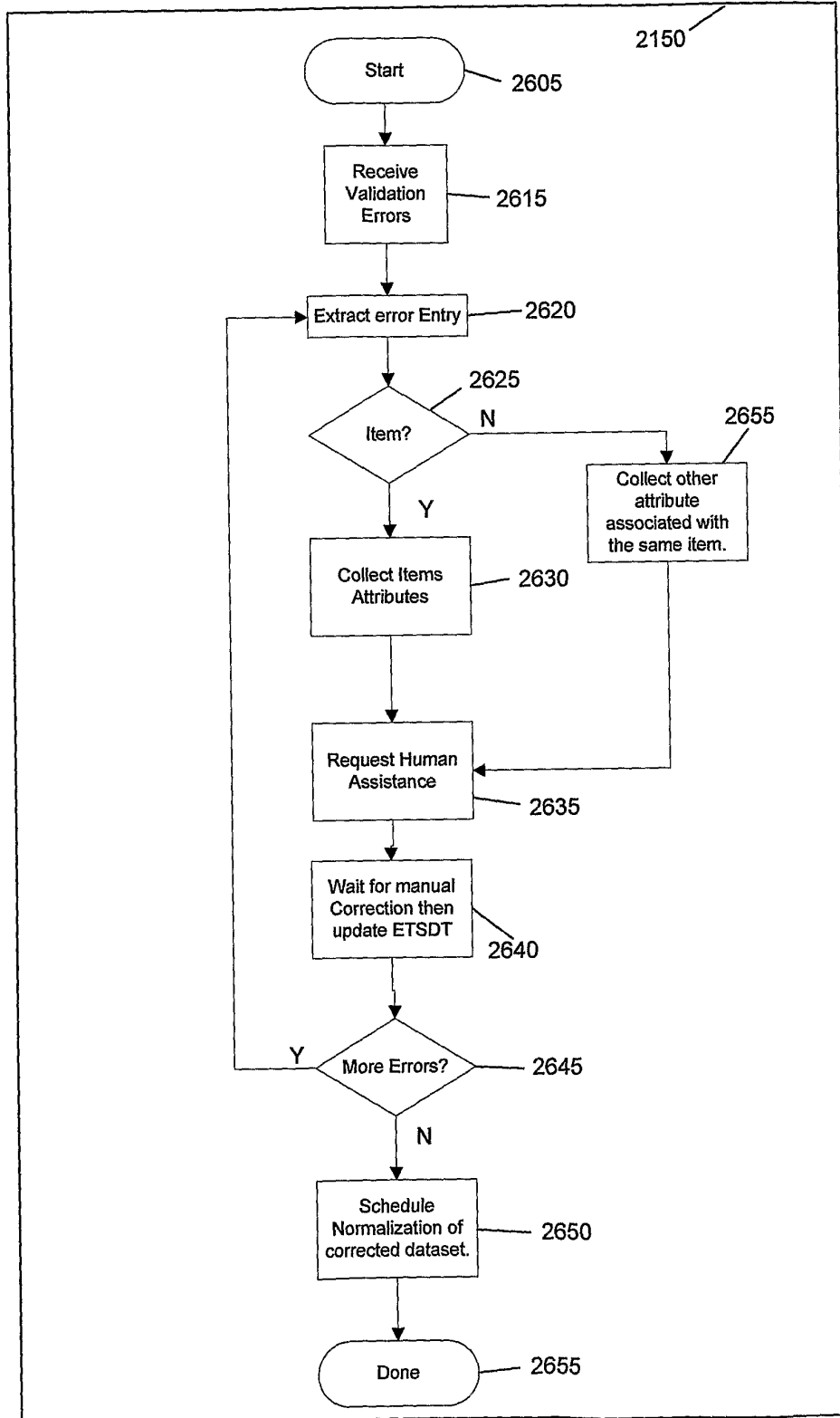
34/47

Figure 16



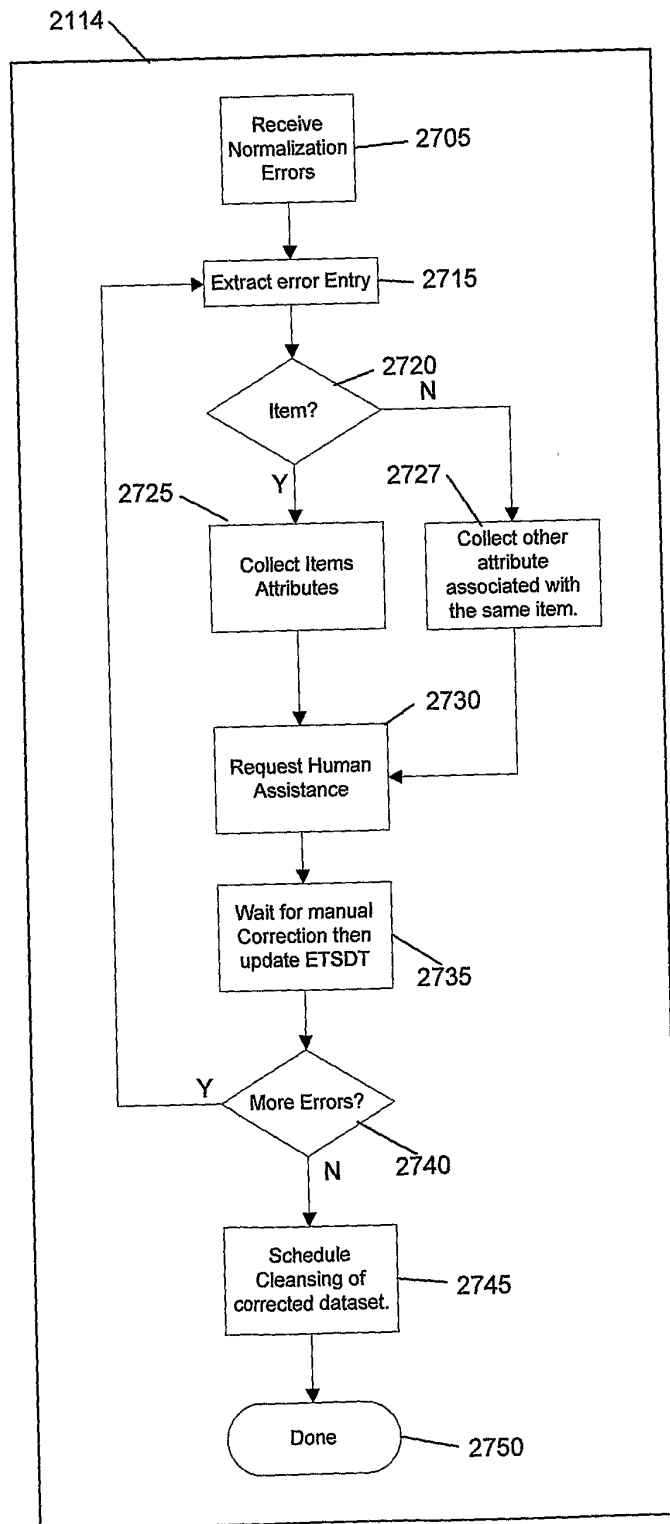
35/47

Figure 17



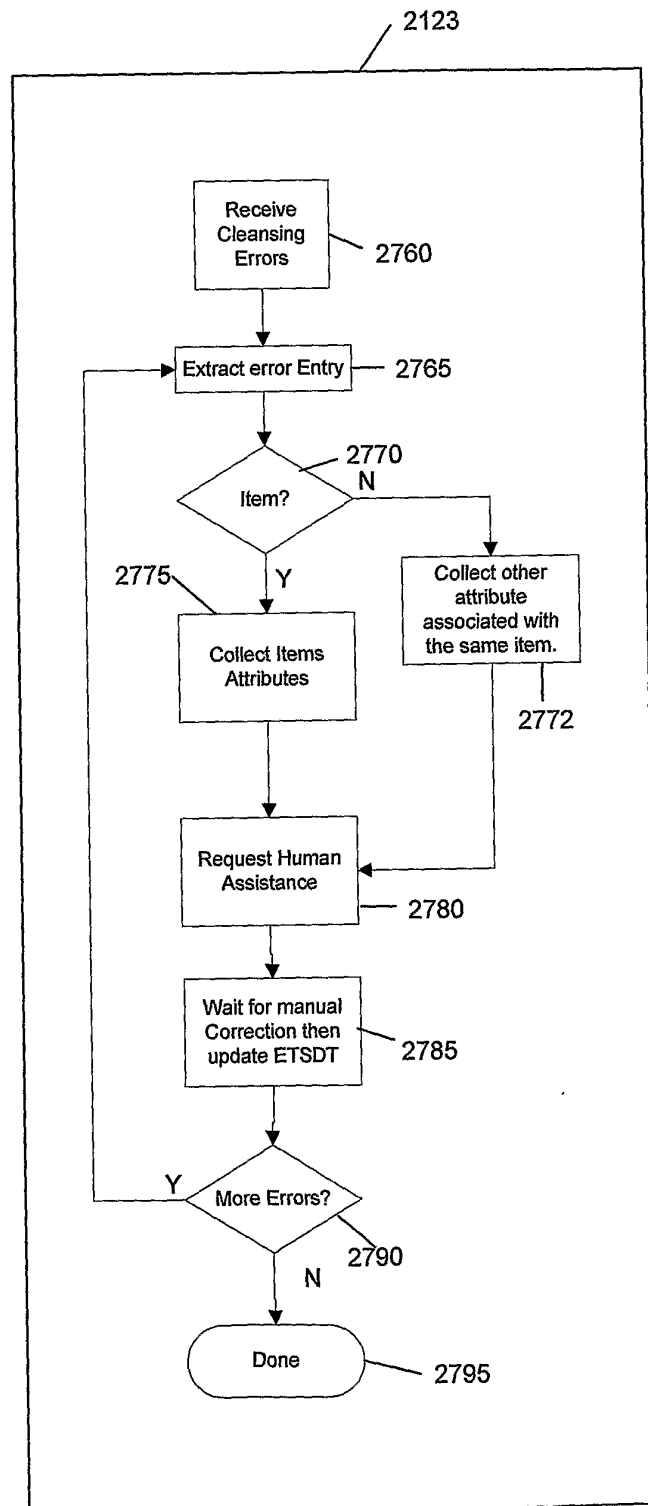
36/47

Figure 18 A



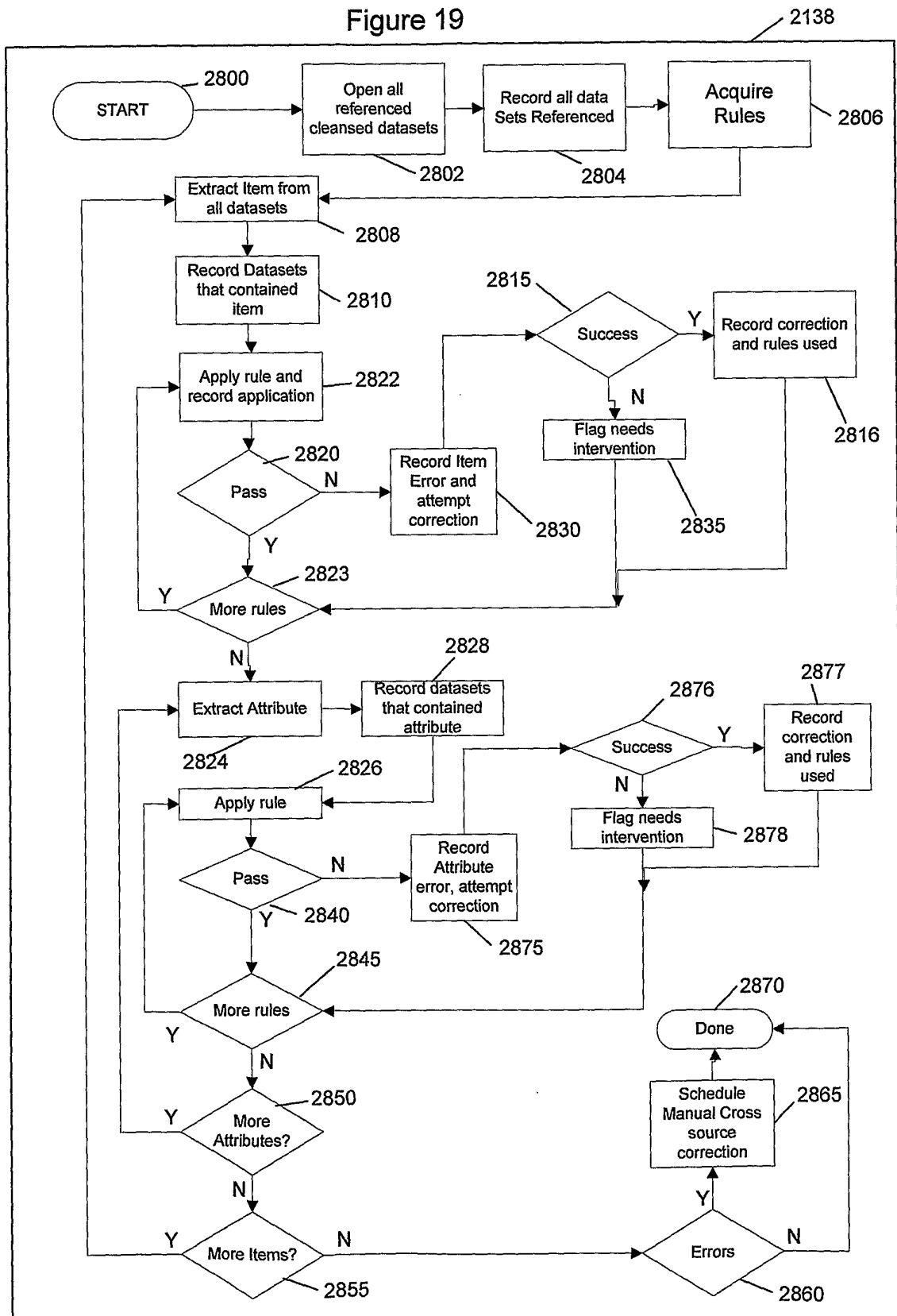
37/47

Figure 18 B



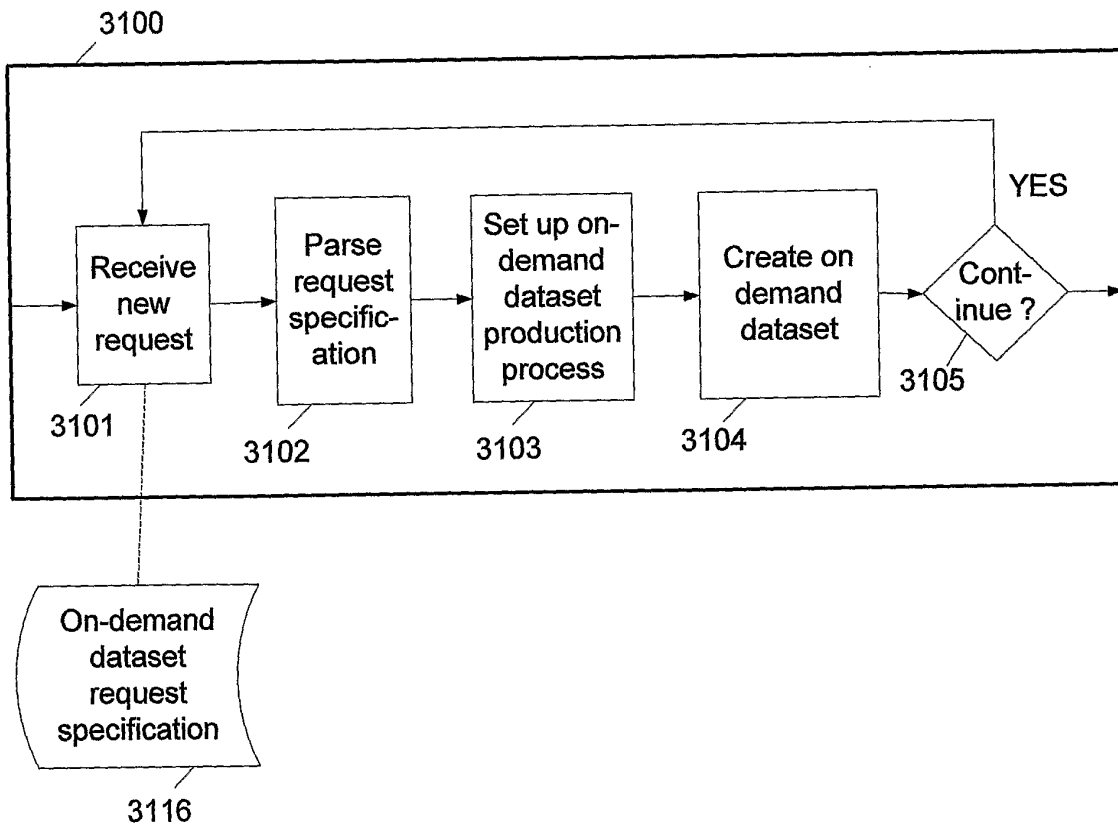
38/47

Figure 19



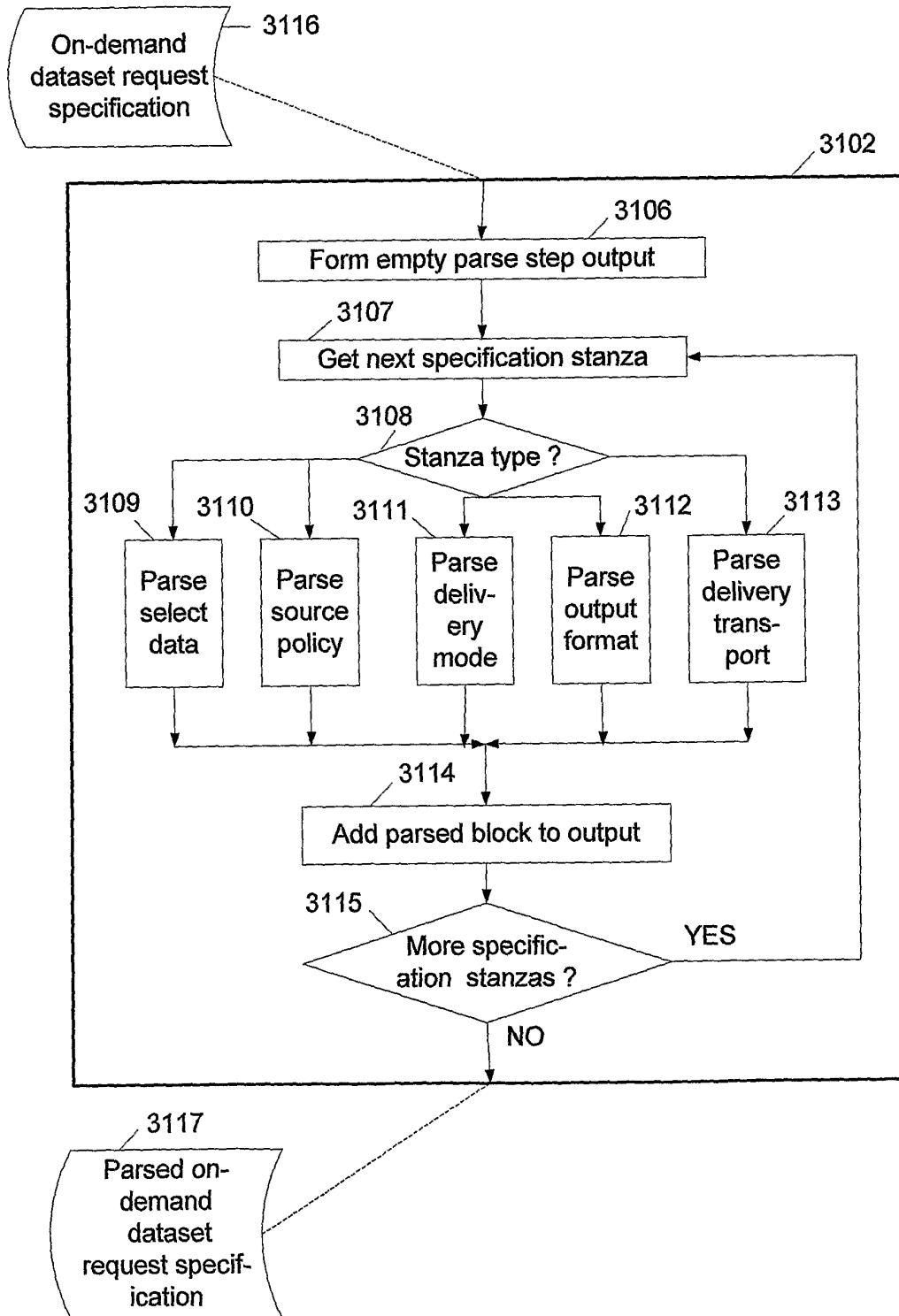
39/47

Figure 20A



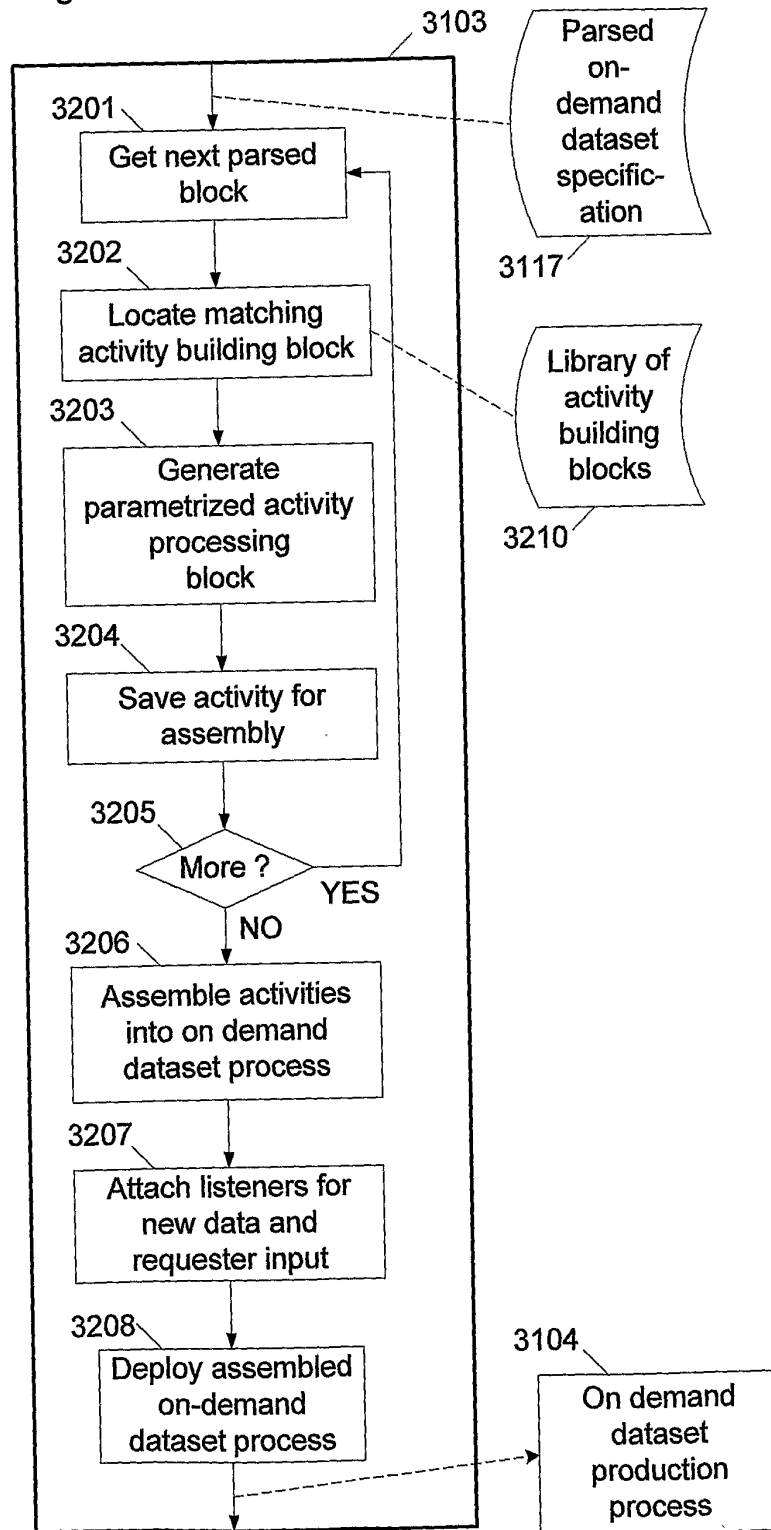
40/47

Figure 20B



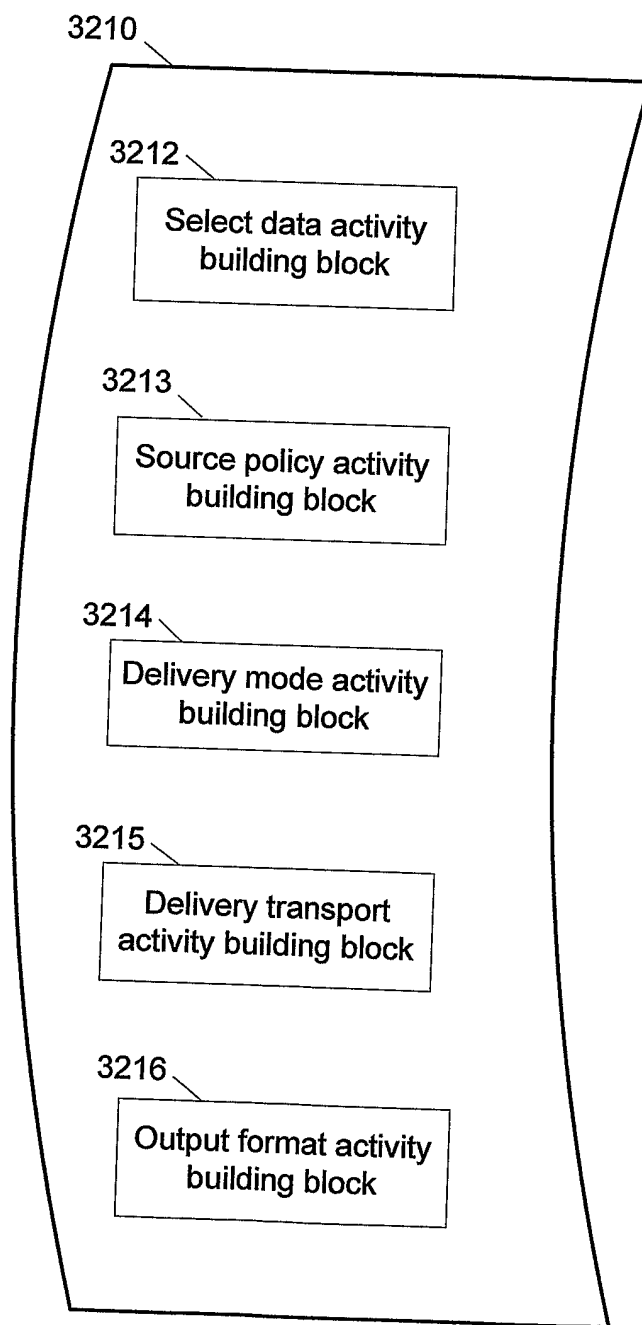
41/47

Figure 21A



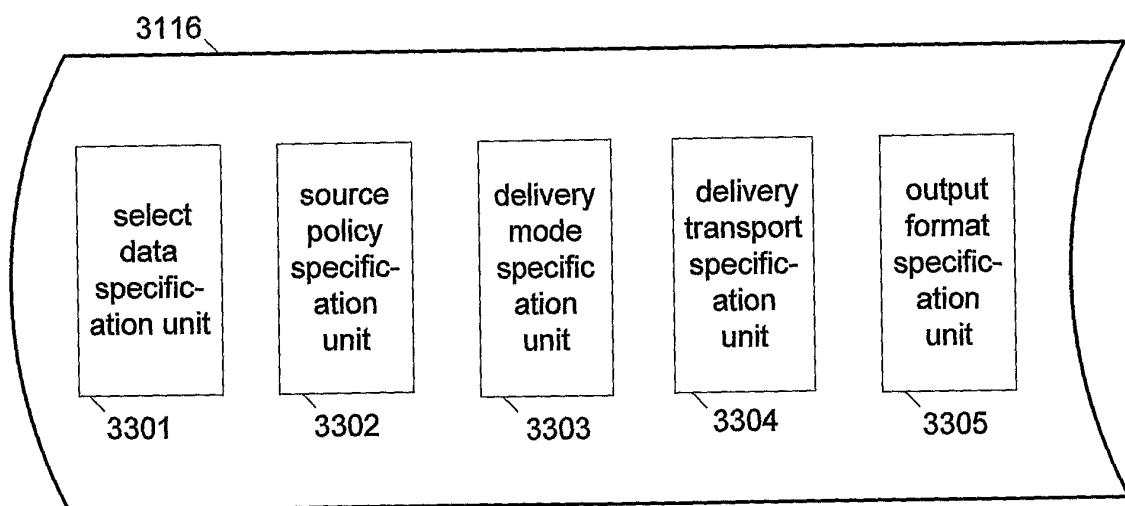
42/47

Figure 21B



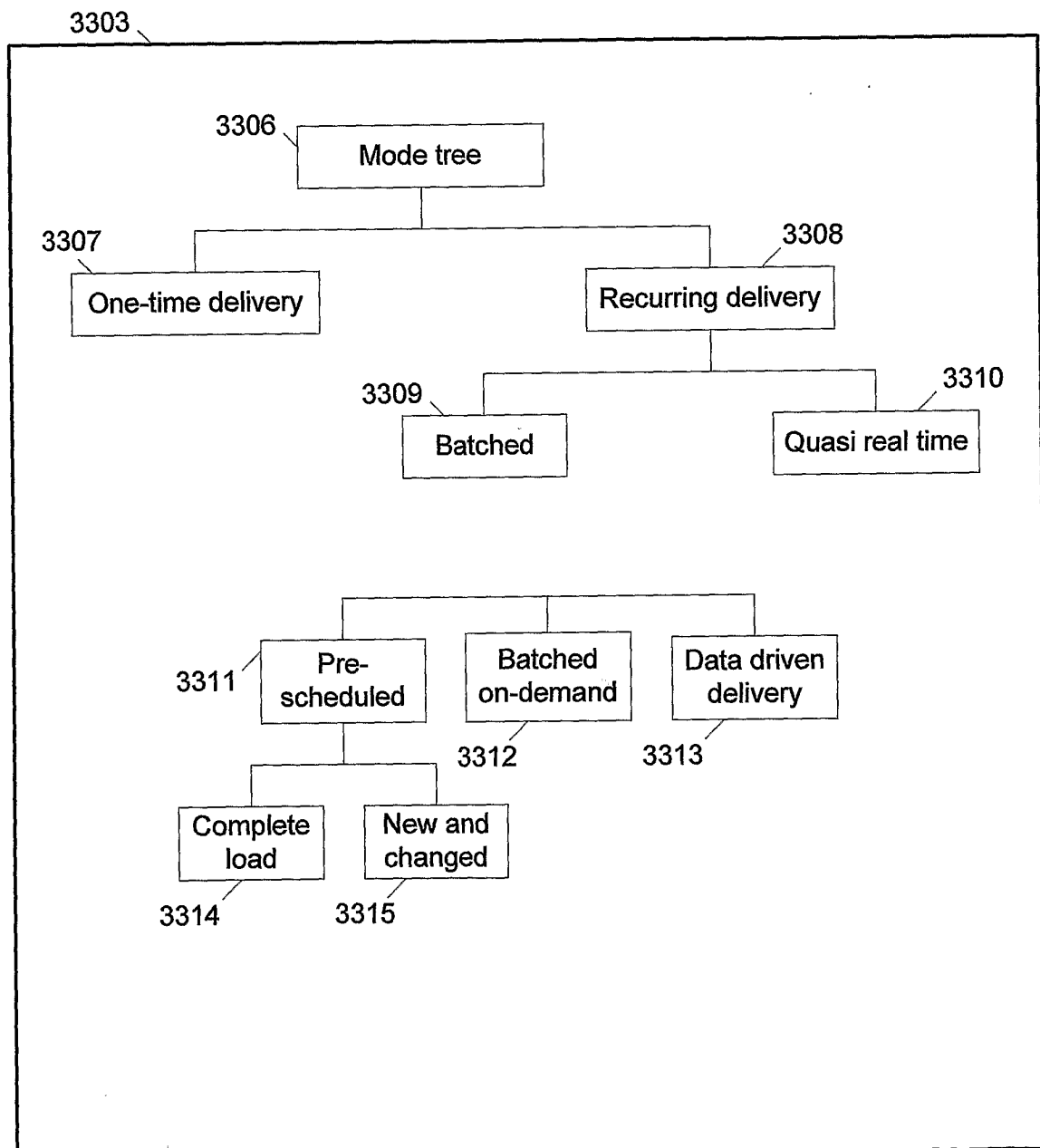
43/47

Figure 22A



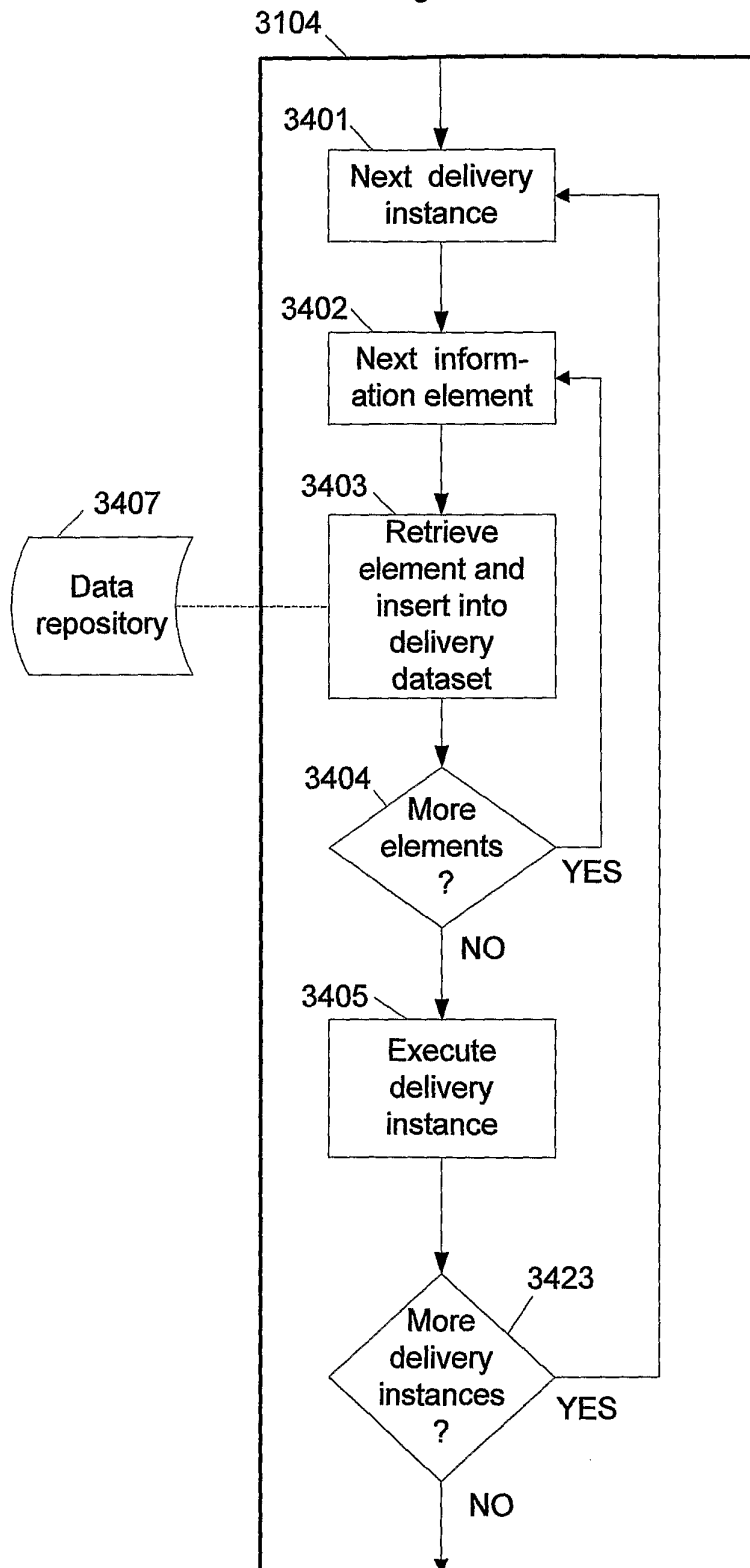
44/47

Figure 22B



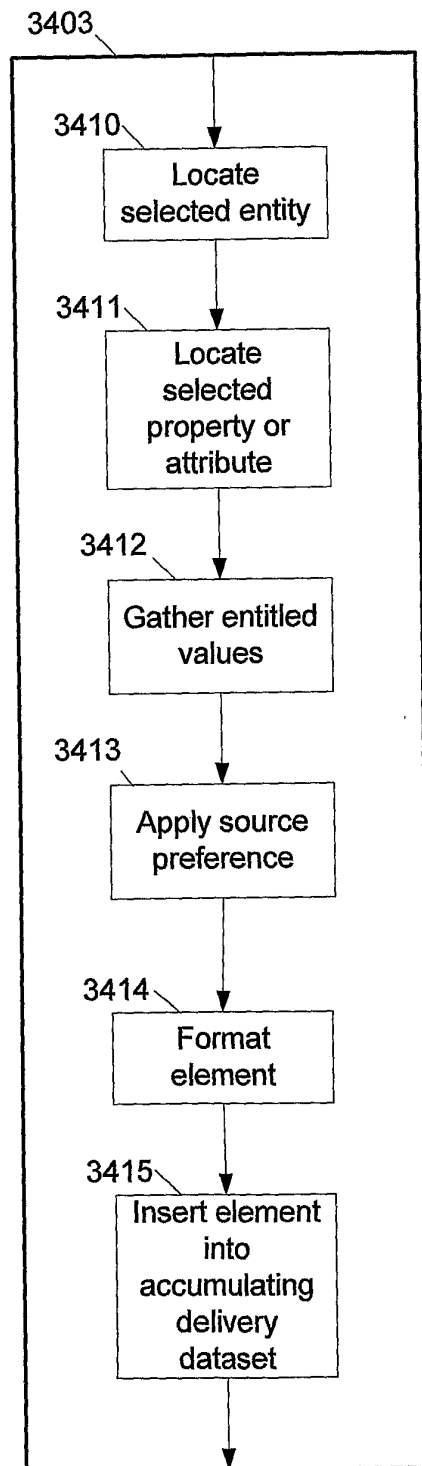
45/47

Figure 23A



46/47

Figure 23B



47/47

Figure 23C

