

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
9 March 2006 (09.03.2006)

PCT

(10) International Publication Number
WO 2006/026702 A2

(51) International Patent Classification:
G06F 15/16 (2006.01)

(21) International Application Number:

PCT/US2005/031097

(22) International Filing Date: 31 August 2005 (31.08.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/606,407 31 August 2004 (31.08.2004) US

(71) Applicant (for all designated States except US): **AS-CENTIAL SOFTWARE CORPORATION** [US/US]; 50 Washington Street, Westborough, MA 01581 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **ANDERSON, Russell, George** [US/US]; 14 Parker Drive, Merrimack, NH 03054 (US). **BOUZIANE, M'hamed** [US/US]; 10 Tempo Drive, Nashua, NH 03062 (US). **MASTRO, Vincent, A.** [US/US]; 62 Greenwood Avenue, Needham, MA 02492 (US). **WEBBER, Robert, C., III** [US/US]; 339 Helmuth Lane, Alexandria, VA 22304 (US).

(74) Agent: **WANG, Elissa, Y.**; International Business Machines Corporation, Intellectual Property Law, SVL, 555 Bailey Avenue #G4, San Jose, CA 95141 (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, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, 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, US, UZ, VC, VN, YU, ZA, ZM, ZW.

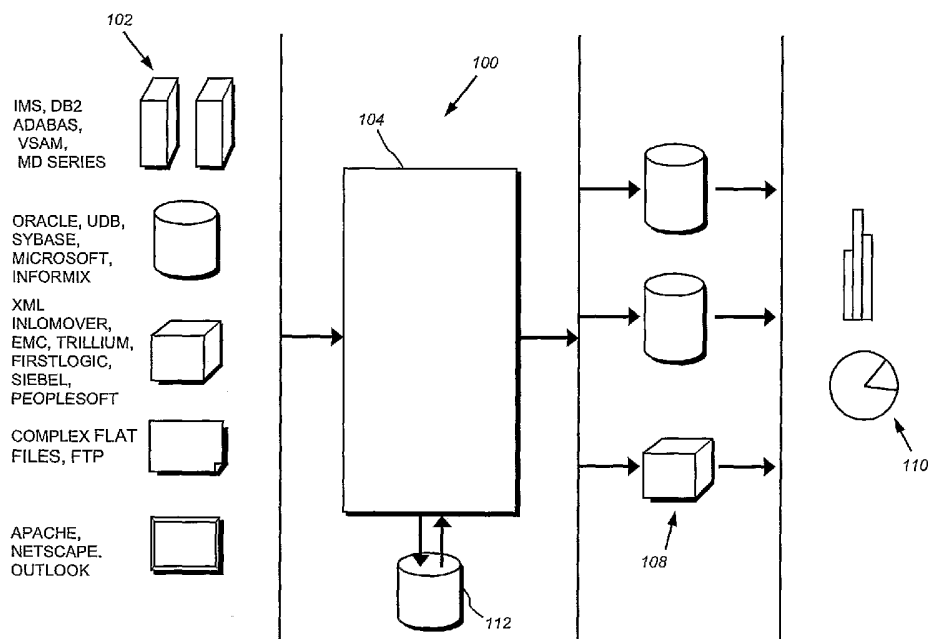
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, 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

[Continued on next page]

(54) Title: METHODS AND SYSTEMS FOR SEMANTIC IDENTIFICATION IN DATA SYSTEMS



(57) Abstract: Provided herein are methods and systems relating to a semantic identifier that may allow for the unique identification of an item based on its relationship with other items without the need to store additional data; a translation engine that may translate data, metadata, semantic identifiers and other items from one format, language and/or data model to another; and a level of abstraction property of a hub or database that may allow for the differentiation of multiple instances or forms of an item.



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.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

A PCT APPLICATION FOR

5

METHODS AND SYSTEMS FOR SEMANTIC IDENTIFICATION IN DATA SYSTEMS

10

15

METHODS AND SYSTEMS FOR SEMANTIC IDENTIFICATION IN DATA SYSTEMS

RELATED APPLICATIONS

5 This application claims priority to U.S. Provisional Application Number 60/606,407, filed August 31, 2004 and entitled "Methods and Systems for Semantic Identification in Data Systems".

BACKGROUND

1. Field.

10 This invention relates to the field of information technology, and more particularly to the field of data integration systems.

2. Description of the Related Art.

15 The advent of computer applications made many business processes much faster and more efficient; however, the proliferation of different computer applications that use different data structures, communication protocols, languages and platforms has led to great complexity in the information technology infrastructure of the typical business enterprise. Different business processes within the typical enterprise may use completely different computer applications, each computer application being developed and optimized for the particular business process, rather than for the enterprise as a whole. For example, a business may have a particular computer application for tracking accounts payable and a completely different one for keeping track of customer contacts. In fact, even the same business process may use more than one computer application, such as when an enterprise keeps a centralized customer contact database, but employees keep their own contact information, such as in a personal information manager.

25 While specialized computer applications offer the advantages of custom-tailored solutions, the proliferation leads to inefficiencies, such as repetitive entry and handling of the same data many times throughout the enterprise, or the failure of the enterprise to capitalize on data that is associated with one process when the enterprise executes another process that could benefit from that data. For example, if the accounts payable process is separated from the supply chain and ordering process, the enterprise may accept and fill orders from a customer whose credit history would have caused the enterprise to decline the order. Many other examples can be provided where an enterprise would benefit from consistent access to all of its data across varied computer applications.

30 A number of companies have recognized and addressed the need for sharing of data across different applications in the business enterprise. Thus, enterprise application integration, or EAI, has emerged as a message-based strategy for addressing data from disparate sources. As computer applications increase in complexity and number, EAI efforts encounter many challenges, ranging from the need to handle different protocols, the need to address ever-increasing volumes of data and numbers of transactions, and an ever-increasing appetite for faster integration of data. Various approaches to EAI have been taken, including least-common-denominator approaches, atomic approaches, and bridge-type approaches. However, EAI is based upon communication between individual applications. As a significant disadvantage, the complexity of EAI solutions grows geometrically in response to linear additions of platforms and applications.

40 While data integration systems provided useful tools for addressing the needs of an enterprise, such systems are typically deployed as custom solutions. They have a lengthy development cycle, and may require sophisticated technical training to accommodate changes in business structure and information requirements. There

remains a need for data integration system tools that permit use, reuse, and modification of functionality in a changing business environment. One such tool is a semantic identifier that may allow for the unique identification of an item based on its relationship with other items without the need to store additional data. A translation engine is another such tool that may translate data, metadata, semantic identifiers and other items from one format, language and/or data model to another. Finally, a level of abstraction property of a hub or database may allow for the differentiation of multiple instances or forms of an item.

SUMMARY

A semantic identifier may exist for an item. The item may be an object, data item, datum, column, row, table, database, instance, attribute, metadata, concept, topic, subject, semantic identifier, other identifier, RFID tag, vendor, supplier, customer, person, team, organization, user, network, system, device, family, store, product, product line, product feature, product specification, product attribute, price, cost, bill of materials, shipping data, tax data, course, educational program, location, map, division, organization, organism, process, rule, law, rating system, good, service and service offering or other item or concept. An item may be related to a data integration job and/or data integration platform. A semantic identifier may identify the item based on the item's relationship with one or more other items. A relationship may be the absence of a relationship. A relationship may be based on semantics. A relationship may involve the position of the item in a relational hierarchy.

A semantic identifier may be a unique identifier for an item. It is possible that a unique semantic identifier for an item takes into account less than all the relationships of that item with other items. It may be advantageous to create a semantic identifier that is based on the minimum number of relationships to ensure uniqueness. The number of relationships required to create a unique semantic identifier for an item may vary based on context. A semantic identifier may be context-dependent. A semantic identifier may be dynamic.

A semantic identifier may be stored, maintained, recorded, processed and/or interpreted in a syntax that may be stored, maintained, recorded, processed and/or interpreted in a string structure or format. The syntax and/or string structure or format may be parseable. The syntax and/or string structure or format may be truncated, modified, shortened, parsed or re-ordered. It may be possible to truncate, modify, shorten or re-order a syntax and/or string and still maintain the unique identifier. A shorter syntax and/or string may be useful in certain contexts and may increase performance.

A semantic identifier may be associated with a semantic context such as a step in an enterprise method, a datum in a database, a datum in a row or column, a row or column in a table, a row or column in a database, a datum in a table, a table in a database, metadata in a database, an item in a hub or repository, an item in a database, an item in a table, an item in a column, an item in a row, a person in an organization, a sender or recipient of a communication, a user on a network, a system on a network, a device on a network, a person in a family, an item in a store, a dish on a menu, a product in a product line, a product in a product offering, a course or step in an educational or training program, a location on a map, a location of an item, a division of an organization, a person on a team, a rule in a system of rules, a service in a service suite, an entity in an organizational hierarchy of an enterprise, an entity in a supply chain, a customer in a market, purchaser in a purchasing decision, a price of a good or service, a cost of a good or service, a component of a product or system, a step of a method and/or a member of a group.

In one embodiment a database may have a table with a column. The unique semantic identifier for that column may be "column name of table name of database name." This unique semantic identifier may be stored,

maintained, recorded, processed and/or interpreted using the following syntax: column name::table name::database name. The syntax and/or any associated string may be parsed and unnecessary elements may be removed. For example, if only one database existed the following syntax may still generate a unique identifier for the column: column name::table name. The database relationship is not required to create a unique semantic identifier. In another example, the database may have only one table, so that the following syntax may be a unique identifier for the column: column name::database name. The table relationship is not required to create a unique identifier. Use of a shorter syntax and/or string may decrease processing times and increase efficiency.

A translation engine may perform translation operations one or more semantic identifiers, databases, databases including semantic identifiers, systems of information, systems of information including semantic identifiers or other items. The translation operation may translate or otherwise modify the format, language and/or data model of a semantic identifier. A translation operation may involve a translation or mapping to or from one or more data tools, languages, formats and/or data models to or from at least one other data tool, language, format and/or data model. A translation operation may involve a translation or mapping to or from DataStage 7, QualityStage, Business Objects, IBM - DB2 Cube Views, UML 1.1, UML 1.3, ERStudio, ProfileStage, . PowerDesigner (with added support for Packages and Extended Attributes) and/or MicroStrategy. A translation engine and/or translation operation may be embodied in a metabroker. A translation engine, a mapping of a translation operation or a translation operation can trace data that is translated in the execution of the operation backward and forward between an original semantic context and a translated semantic context. A translation operation may be performed, executed and/or conducted in batch, real-time and/or on a continuous basis. A translation operation may be provided or made available as a service, for example, as part of a service oriented architecture.

Once a translation operation exists for a semantic identifier, database, database including one or more semantic identifiers, system of information, system of information including one or more semantic identifiers or other item it can be translated to or from, mapped to, linked to, used with or associated with any other semantic identifier, database, database including one or more semantic identifiers, system of information, system of information including one or more semantic identifiers or other item sharing at least one translation operation.

An item may exist in multiple forms or instances, such as a physical modeling activity and/or logical modeling activity. An item, including any associated data or metadata, may exist in multiple forms or instances in a database and/or hub. In order to distinguish between the various forms or instances of an item, any differentiating characteristic may be used, such as a level of abstraction, a position in a hierarchy, a relationship to another item, one or more distinguishing attributes of the item, the context in which the item is found, the physical location in which the item is found, or the like.

In one embodiment, a table named "employee" may be brought into the hub. The hub collector may have two forms or instances of "employee" in the hub; one corresponding to the physical modeling activity and another corresponding to the logical modeling activity. The level of abstraction property of hub data collection allows for the differentiation between the physical model and logical model instances or forms.

When performing a translation operation, which may be in response to a query, a translation engine may grab, load or obtain all of the items from a hub or database. It may then filter, select, store, translate, modify, or otherwise operate on the items based on a distinguishing characteristic, such as abstraction level, position in a hierarchy, a relationship to another item, an attribute of the items, a physical locations or the like. In the alternative, when performing a translation operation, which may be in response to a query, a translation engine may filter,

select, store, translate, modify or otherwise operate on items, including any data and/or metadata, at the hub or database and grab, load or obtain only those items of the relevant level of abstraction or having the relevant attributes, positions, relationships, locations or the like. The filtering, selection, storage, translation, modification or other operation may be performed at runtime or design time and may be conducted in batch, real-time or on a continuous basis. In embodiments, the filtering, selection, storage, translation, modification or other operation may be based on information or inputs obtained by the translation engine and/or system at development-time, design-time or run-time, such as data model, a mapping of a data model, a differentiating characteristic of the syntax of an identifier, or the like. The information may be updated in a dynamic fashion in real-time. Thus, in one preferred embodiment, a system may refine a select command for selecting data from a database based on a known mapping of the database, such as to select logical items and omit physical items, or vice versa. A query may be a message or operation.

In some cases the closer in the overall process the filtering, selection or other operation is to the hub or database, the more efficient and faster the operation. The translation engine may perform a translation operation on the query itself resulting in a revised query or select command which may be sent directly to the hub or database. The revised query or select command may be in a format directly compatible with the hub or database.

In other aspects, a computer program product may include a computer useable medium including computer readable program code, wherein the computer readable program code when executed on one or more computers causes the one or more computers to perform any one or more of the methods above.

"International Business Machines" or "IBM" as used herein shall refer to International Business Machines Corporation of Armonk, New York.

As used herein, "data source" or "data target" are intended to have the broadest possible meaning consistent with these terms, and shall include a database, a plurality of databases, a repository information manager, a queue, a message service, a repository, a data facility, a data storage facility, a data provider, a website, a server, a computer, a computer storage facility, a CD, a DVD, a mobile storage facility, a central storage facility, a hard disk, a multiple coordinating data storage facilities, RAM, ROM, flash memory, a memory card, a temporary memory facility, a permanent memory facility, magnetic tape, a locally connected computing facility, a remotely connected computing facility, a wireless facility, a wired facility, a mobile facility, a central facility, a web browser, a client, a laptop, a personal digital assistant ("PDA"), a telephone, a cellular phone, a mobile phone, an information platform, an analysis facility, a processing facility, a business enterprise system or other facility where data is handled or other facility provided to store data or other information, as well as any files or file types for maintaining structured or unstructured data used in any of the above systems, or any streaming, messaged, event driven, or otherwise sourced data, and any combinations of the foregoing, unless a specific meaning is otherwise indicated or the context of the phrase requires otherwise. A storage mechanism is any logical or physical device, resource, or facility capable of acting as a data source or data target.

"Enterprise Java Bean (EJB)" shall include the server-side component architecture for the J2EE platform. EJBs support rapid and simplified development of distributed, transactional, secure and portable Java applications. EJBs support a container architecture that allows concurrent consumption of messages and provide support for distributed transactions, so that database updates, message processing, and connections to enterprise systems using the J2EE architecture can participate in the same transaction context.

"JMS" shall mean the Java Message Service, which is an enterprise message service for the Java-based J2EE enterprise architecture. "JCA" shall mean the J2EE Connector Architecture of the J2EE platform described

more particularly below. It should be appreciated that, while EJB, JMS, and JCA are commonly used software tools in contemporary distributed transaction environments, any platform, system, or architecture providing similar functionality may be employed with the data integration systems described herein.

5 “Real time” as used herein, shall include periods of time that approximate the duration of a business transaction or business and shall include processes or services that occur during a business operation or business process, as opposed to occurring off-line, such as in a nightly batch processing operation. Depending on the duration of the business process, real time might include seconds, fractions of seconds, minutes, hours, or even days.

10 “Business process,” “business logic” and “business transaction” as used herein, shall include any methods, service, operations, processes or transactions that can be performed by a business, including, without limitation, sales, marketing, fulfillment, inventory management, pricing, product design, professional services, financial services, administration, finance, underwriting, analysis, contracting, information technology services, data storage, data mining, delivery of information, routing of goods, scheduling, communications, investments, transactions, offerings, promotions, advertisements, offers, engineering, manufacturing, supply chain management, human
15 resources management, data processing, data integration, work flow administration, software production, hardware production, development of new products, research, development, strategy functions, quality control and assurance, packaging, logistics, customer relationship management, handling rebates and returns, customer support, product maintenance, telemarketing, corporate communications, investor relations, and many others.

20 “Service oriented architecture (SOA)”, as used herein, shall include services that form part of the infrastructure of a business enterprise. In the SOA, services can become building blocks for application development and deployment, allowing rapid application development and avoiding redundant code. Each service may embody a set of business logic or business rules that can be bound to the surrounding environment, such as the source of the data inputs for the service or the targets for the data outputs of the service. Various instances of SOA are provided in the following description.

25 “Metadata,” as used herein, shall include data that brings context to the data being processed, data about the data, information pertaining to the context of related information, information pertaining to the origin of data, information pertaining to the location of data, information pertaining to the meaning of data, information pertaining to the age of data, information pertaining to the heading of data, information pertaining to the units of data, information pertaining to the field of data and/or information pertaining to any other information relating to the
30 context of the data.

“WSDL” or “Web Services Description Language” as used herein, includes an XML format for describing network services (often web services) as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined
35 into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate.

“Metabroker” as used herein, shall include systems or methods that may involve a translation engine or other means for performing translation operations or other operations on data or metadata. The translation operations or other operations may involve the translation of data or metadata from one or more formats, languages
40 and/or data models to one or more formats, languages and/or data models.

BRIEF DESCRIPTION OF THE FIGURES

Fig. 1 is a schematic diagram of a business enterprise with a plurality of business processes, each of which may include a plurality of different computer applications and data sources.

Fig. 2 is a schematic diagram showing data integration across a plurality of business processes of a business enterprise.

Fig. 3 is a schematic diagram showing an architecture for providing data integration for a plurality of data sources for a business enterprise.

Fig. 4 shows an item in relation to other items.

Fig. 5 shows an item in relation to other items.

Fig. 6A shows an item in a certain context.

Fig. 6B shows an item in a certain context.

Fig. 7 shows certain strings.

Fig. 8 shows an item and a corresponding string.

Fig. 9 shows a string and certain of its variations.

Fig. 10 shows a translation engine acting on certain strings.

Fig. 11 shows an item that may exist in multiple forms or instances.

Fig. 12 shows an item that may exist in multiple forms or instances in a hub or database.

Fig. 13 shows an item in a hub at various levels of abstraction.

Fig. 14 shows a translation process in which all items are grabbed at the database or hub.

Fig. 15 shows a translation process in which items are filtered at the database or hub.

Fig. 16 shows a translation process in which the query is translated.

DETAILED DESCRIPTION

Throughout the following discussion, like element numerals are intended to refer to like elements, unless specifically indicated otherwise.

The invention(s) disclosed herein can take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment containing both hardware and software elements. In a preferred embodiment, the invention is implemented in software, which includes but is not limited to firmware, resident software, microcode, etc.

Furthermore, the invention(s) can take the form of a computer program product accessible from a computer-usable or computer-readable medium providing program code for use by or in connection with a computer or any instruction execution system. For the purposes of this description, a computer-usable or computer-readable medium can be any apparatus that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device.

The medium can be an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system (or apparatus or device) or a propagation medium. Examples of a computer-readable medium include a semiconductor or solid state memory, magnetic tape, a removable computer diskette, a random access memory (RAM), a read-only memory (ROM), a rigid magnetic disk and an optical disk. Current examples of optical disks include compact disk – read only memory (CD-ROM), compact disk – read/write (CD-R/W) and DVD.

A data processing system suitable for storing and/or executing program code will include at least one processor coupled directly or indirectly to memory elements through a system bus. The memory elements can

include local memory employed during actual execution of the program code, bulk storage, and cache memories which provide temporary storage of at least some program code in order to reduce the number of times code must be retrieved from bulk storage during execution.

Input/output or I/O devices (including but not limited to keyboards, displays, pointing devices, etc.) can be coupled to the system either directly or through intervening I/O controllers.

Network adapters may also be coupled to the system to enable the data processing system to become coupled to other data processing systems or remote printers or storage devices through intervening private or public networks. Modems, cable modem and Ethernet cards are just a few of the currently available types of network adapters.

Fig. 1 represents a platform 100 for facilitating integration of various data of a business enterprise. The platform includes a plurality of business processes, each of which may include a plurality of different computer applications and data sources. The platform may include several data sources 102, which may be data sources such as those described above. These data sources may include a wide variety of data types from a wide variety of physical locations. For example, the data source may include systems from providers such as Sybase, Microsoft, Informix, Oracle, Inlomover, EMC, Trillium, First Logic, Siebel, PeopleSoft, IBM, Apache, or Netscape. The data sources 102 may include systems using database products or standards such as IMS, DB2, ADABAS, VSAM, MD Series, UDB, XML, complex flat files, or FTP files. The data sources 102 may include files created or used by applications such as Microsoft Outlook, Microsoft Word, Microsoft Excel, Microsoft Access, as well as files in standard formats such as ASCII, CSV, GIF, TIF, PNG, PDF, and so forth. The data sources 102 may come from various locations or they may be centrally located. The data supplied from the data sources 102 may come in various forms and have different formats that may or may not be compatible with one another.

Data targets are discussed later in this description. In general, these data targets may be any of the data sources 102 noted above. This difference in nomenclature typically denotes whether a data system provides data or receives data in a data integration process. However, it should be appreciated that this distinction is not intended to convey any difference in capability between data sources and data targets (unless specifically stated otherwise), since in a conventional data integration system, data sources may receive data and data targets may provide data.

The platform illustrated in Fig.1 also includes a data integration system 104. The data integration system may, for example, facilitate the collection of data from the data sources 102 as the result of a query or retrieval command the data integration system 104 receives. The data integration system 104 may send commands to one or more of the data sources 102 such that the data source(s) provides data to the data integration system 104. Since the data received may be in multiple formats including varying metadata, the data integration system may reconfigure the received data such that it can be later combined for integrated processing. The functions that may be performed by the data integration system 104 are described in more detail below.

The platform 100 also includes several retrieval systems 108. The retrieval systems 108 may include databases or processing platforms used to further manipulate the data communicated from the data integration system 104. For example, the data integration system 104 may cleanse, combine, transform or otherwise manipulate the data it receives from the data sources 102 such that a retrieval system 108 can use the processed data to produce reports 110 useful to the business. The reports 110 may be used to report data associations, answer complex queries, answer simple queries, or form other reports useful to the business or user, and may include raw data, tables, charts, graphs, and any other representations of data from the retrieval systems 108.

The platform 100 may also include a database or data base management system 112. The database 112 may be used to store information temporally, temporarily, or for permanent or long-term storage. For example, the data integration system 104 may collect data from one or more data sources 102 and transform the data into forms that are compatible with one another or compatible to be combined with one another. Once the data is transformed, the data integration system 104 may store the data in the database 112 in a decomposed form, combined form or other form for later retrieval.

Fig. 2 is a schematic diagram showing data integration across a plurality of entities and business processes of a business enterprise. In the illustrated embodiment, the data integration system 104 facilitates the information flowing between user interface systems 202 and data sources 102. The data integration system 104 may receive queries from the interface systems 202, where the queries necessitate the extraction and possibly transformation of data residing in one or more of the data sources 102. The interface systems 202 may include any device or program for communicating with the data integration system 104, such as a web browser operating on a laptop or desktop computer, a cell phone, a personal digital assistant ("PDA"), a networked platform and devices attached thereto, or any other device or system that might interface with the data integration system 104.

For example, a user may be operating a PDA and make a request for information to the data integration system 104 over a WiFi or Wireless Access Protocol/Wireless Markup Language ("WAP/WML") interface. The data integration system 104 may receive the request and generate any required queries to access information from a website or other data source 102 such as an FTP file site. The data from the data sources 102 may be extracted and transformed into a format compatible with the requesting interface system 202 (a PDA in this example) and then communicated to the interface system 202 for user viewing and manipulation. In another embodiment, the data may have previously been extracted from the data sources and stored in a separate database 112, which may be a data warehouse or other data facility used by the data integration system 104. The data may have been stored in the database 112 in a transformed condition or in its original state. For example, the data may be stored in a transformed condition such that the data from a number of data sources 102 can be combined in another transformation process. For example, a query from the PDA may be transmitted to the data integration system 104 and the data integration system 104 may extract the information from the database 112. Following the extraction, the data integration system 104 may transform the data into a combined format compatible with the PDA before transmission to the PDA.

Fig. 3 is a schematic diagram showing an architecture for providing data integration for a plurality of data sources 102 for a business enterprise. An embodiment of a data integration system 104 may include a discover data stage 302 to perform, possibly among other processes, extraction of data from a data source and analysis of column values and table structures for source data. A discover data stage 302 may also generate recommendations about table structure, relationships, and keys for a data target. More sophisticated profiling and auditing functions may include date range validation, accuracy of computations, accuracy of if-then evaluations, and so forth. The discover data stage 302 may normalize data, such as by eliminating redundant dependencies and other anomalies in the source data. The discover data stage 302 may provide additional functions, such as drill down to exceptions within a data source 102 for further analysis, or enabling direct profiling of mainframe data. A non-limiting example of a commercial embodiment of a discover data stage 302 may be found in IBM's WebSphere ProfileStage product.

The data integration system 104 may also include a data preparation stage 304 where the data is prepared, standardized, matched, or otherwise manipulated to produce quality data to be later transformed. The data preparation stage 304 may perform generic data quality functions, such as reconciling inconsistencies or checking

for correct matches (including one-to-one matches, one-to-many matches, and deduplication) within data. The data preparation stage 304 may also provide specific data enhancement functions. For example, the data preparation stage 304 may ensure that addresses conform to multinational postal references for improved international communication. The data preparation stage 304 may conform location data to multinational geocoding standards for spatial information management. The data preparation stage may modify or add to addresses to ensure that address information qualifies for U.S. Postal Service mail rate discounts under Government Certified U.S. Address Correction. Similar analysis and data revision may be provided for Canadian and Australian postal systems, which provide discount rates for properly addressed mail. A non-limiting example of a commercial embodiment of a data preparation stage 304 may be found in IBM's WebSphere QualityStage product.

The data integration system may also include a data transformation stage 308 to transform, enrich and deliver transformed data. The data transformation stage 308 may perform transitional services such as reorganization and reformatting of data, and perform calculations based on business rules and algorithms of the system user. The data transformation stage 308 may also organize target data into subsets known as datamarts or cubes for more highly tuned processing of data in certain analytical contexts. The data transformation stage 308 may employ bridges, translators, or other interfaces (as discussed generally below) to span various software and hardware architectures of various data sources and data targets used by the data integration system 104. The data transformation stage 308 may include a graphical user interface, a command line interface, or some combination of these, to design data integration jobs across the platform 100. A non-limiting example of a commercial embodiment of a data transformation stage 308 may be found in IBM's WebSphere DataStage product.

The stages 302, 304, 308 of the data integration system 104 may be executed using a parallel execution system 310 or in a serial or combination manner to optimize the performance of the system 104.

The data integration system 104 may also include a metadata management system 312 for managing metadata associated with data sources 102. In general, the metadata management system 312 may provide for interchange, integration, management, and analysis of metadata across all of the tools in a data integration environment. For example, a metadata management system 312 may provide common, universally accessible views of data in disparate sources, such as IBM's WebSphere ODBC MetaBroker, CA ERwin, IBM's WebSphere ProfileStage, IBM's WebSphere DataStage, IBM's WebSphere QualityStage, IBM DB2 Cube Views, and Cognos Impromptu. The metadata management system 312 may also provide analysis tools for data lineage and impact analysis for changes to data structures. The metadata management system 312 may further be used to prepare a business data glossary of data definitions, algorithms, and business contexts for data within the data integration system 104, which glossary may be published for use throughout an enterprise. A non-limiting example of a commercial embodiment of a metadata management system 312 may be found in IBM's WebSphere MetaStage product.

Referring to Fig. 4, items that are relevant to an enterprise can be described in terms of various contexts or hierarchies, such as to capture the semantic context of the items. Thus, Fig. 4 depicts a semantic identifier for an item. The item may be an object, class, attribute, data item, data model, metadata model, model, definition, identity, structure, language, mapping, relationship, instance or other item or concept, including another semantic identifier. The semantic identifier may identify the item based on the item's attributes, the item's physical location, the relationship of the item with one or more other items, such as in a hierarchy, or the like. In some cases a relationship may be defined as the absence of some particular relationship. A relationship may be based on semantics. A relationship may involve the position of the item in a relational hierarchy. For example, in Fig. 4 item

1 5202 may be identified based on its relationship with the other items to which it is related. Item 1 5202 may be identified as being directly related to item 2 5204, item 3 5208 and item 4 5210, indirectly related to item 5 5212 and indirectly related to item 6 5214 through item 5 5212 and item 4 5210. Item 1 may also be identified as being directly related to item 2 5204, item 3 5208 and item 4 5210. In embodiments, the indirect relationships between item 1 5202 and item 5 5212 and item 6 5214 may be captured in the relationship of item 5202 1 to item 4 5210. This concatenation or recursive type of identification may permit dynamic, in addition to static, identifiers. For example, if the relationship between item 4 5210 and item 6 5214 changes, the semantic identifier for item 1 5202 which incorporates item 2 5204, item 3 5208 and item 4 5210 would incorporate this change through incorporation of item 4 5210 and would not need to be updated to account for the changes in item 6 5214 as it would if item 6 5214 was directly included in the semantic identifier.

Figure 5 presents a more concrete example of a semantic identifier. Jim may be identified as Jim, residing at 111 Anyroad, Anytown, Anystate USA, with phone number 555-555-5555 and social security number 013-65-8067. Alternatively, Jim may be identified in terms of his relationships with others. As depicted in Figure 5, Jim may be identified as the son of Betty, brother of Larry and Jeff, father of Jessica and nephew of Frank.

The semantic identifier may be a unique identifier for an item. In the example of Figure 5, if there were only one Jim in the world who was the son of Betty, brother of Larry and Jeff, father of Jessica and nephew of Frank, this semantic identifier would be a unique identifier for Jim. It is possible that a unique semantic identifier to an item takes into account fewer than all of the relationships of that item with other items. In the example of Figure 5, if there were only one Jim in the world who was the son of Betty, brother of Larry and father of Jessica, the existence of these relationships alone would be enough to create a unique semantic identifier. Jim's relationships with Jeff and Frank would not need to be considered. It may be advantageous to create a semantic identifier that is based on the minimum number of relationships that ensure uniqueness. For example, if the semantic identifier was to be stored in a database 112 or processed by a data integration system 104, a less complex semantic identifier would require less space and would allow for faster processing.

The number of relationships required to create a unique semantic identifier for an item may vary based on context. Figure 6A depicts two items of interest: item 1 5402 and item 7 5404. In context A 5408, item 1 5402 may be distinguished from item 7 5404 by item 1's 5402 relationship with item 5 5410 and item 6 5412. That is, in context A, the unique semantic identifier for item 1 5402 may be that it is directly related to items 2, 3 and 4, indirectly related to item 5 5410 through item 4 and indirectly related to item 6 5412 through item 5 5410 and item 4. In context A, the unique semantic identifier for item 7 5404 may be that it is directly related to only items 2 and 3. Figure 6B presents item 1 5402 in a different context, context B 5414. To uniquely identify item 1 5402 in context B 5414 any one or more of item 1's 5402 direct relationships with item 4, absence of a direct relationship with item 6 or indirect relationship with item 5 may be taken into account. In context B 5414 item 1 5402 may be uniquely semantically identified as directly related to items 2 and 3, but not directly related to item 6. Thus, the unique identifier for item 1 differs between context A 5408 and context B 5414. Thus, in embodiments of the data integration methods and systems described herein, a semantic identifier for an item, such as an item related to a data integration job or a data integration platform, may be provided with a context-dependent identifier for the item. In embodiments such a context-dependent identifier may be stored in an atomic format, such as in a data repository.

In other embodiments, contexts A 5408 and B 5414 may be two different imports, mappings, run versions, models, metabroker models, instances, tools, views, objects, classes, items, relationships, attributes, or any combination of any of the foregoing. A matching or comparison facility may compare the syntax of the identity of

an item in different imports, run versions, models, metabroker models, instances, tools and/or items and determine or assist with the determination of what action to take or refrain from taking based on the comparison. For example, a matching engine may compare the model used by import instance A to the model used by metabroker B. Based on this comparison it may be decided that metabroker B can access the data and metadata of import instance
 5 A without transformation or modification, and the comparison facility may direct the metabroker B to proceed. In another example, tool A 5408 may be compared to tool B 5414, and it may be determined to perform a cross-tool object merge, wherein each tool can access and use the objects of the other tool. In embodiments the comparison facility may trigger a translation facility to assist the cross-tool object merge, such as establishing a bridge, metabroker, hub or the like for translating any objects that require translation, such as translation that is based on
 10 the different syntax for the handling of the identity of particular items in each respective tool, or based on other differences between the tools as determined by the comparison.

In embodiments a semantic identifier may be stored, maintained, recorded, processed and/or interpreted in a syntax that may be stored, maintained, recorded, processed and/or interpreted in a string structure or format. Figure 7 depicts an example of a syntax and a corresponding string composed in that syntax. The syntax 5502 may
 15 be column name::table name::database name. This syntax may be related, for example, to a semantic identifier that identifies a column of a table in a database. A string composed in this syntax 5504 may be age::employee::employee database. This string may be related, for example, to a semantic identifier that identifies the age of an employee in a particular employee database. In the example of Figure 6B, the string corresponding to the semantic identifier for item 1 5402 in context B 5414 may be: direct relation to item 2::direct relation to item
 20 3::direction relationship to item 4. The semantic identifier and corresponding string may also incorporate the lack of a direct relationship between items 1 5402 and item 6.

In Figure 8 the semantic identifier in string format for item 9 5602 may be: direct to item 2::direct to item 3::direct to item 4::indirect to item 5 5604. A string may be capable of being parsed. A syntax and/or string may be truncated, modified and/or the elements of a syntax and/or string may be re-ordered. In Figure 9 string 5702 is a
 25 truncation of string 5604, string 5704 is a truncation and modification and/or re-ordering of string 5604 and string 5708 is a modification and/or re-ordering of string 5606. The truncation, modification and/or re-ordering may be performed by a translation engine. It may be useful to truncate a syntax and/or string when all of the relationships included in the syntax and/or string are not required for the uniqueness of the semantic identifier. Suppose that in a given context for string 5604 all items were directly related to item 3; for example, item 3 was a database in which
 30 all the items were stored. String 5604 could be truncated, such as to create string 5702, omitting the relationship-involving item 3, and still remain a unique semantic identifier. Truncating a syntax and/or string may reduce storage requirements and increase processing efficiency. It may also be useful to change the order of the relationships in a syntax and/or string, for example, to reduce processing time for data integration processes. If the less common relationships are processed first, a system will likely need to access and process fewer relationships
 35 associated with an item in order to identify the item. For example, if very few items were related to item 3, even fewer related to item 4 and many items related to item 2, depending on the context, string 5708 may allow for the identification of item 9 in a shorter time than string 5604. It could be that only the first two elements of string 5708 are needed to uniquely identify item 9 in the context, while the first three elements of string 5604 are needed.

A translation engine may perform translation operations with respect to one or more semantic identifiers, databases 112, databases 112 including semantic identifiers, systems of information, systems of information
 40 including semantic identifiers or other items. Figure 10 depicts a translation engine 5802 acting on a semantic

identifier embodied as a string 5804 and on a semantic identifier embodied as a string located in a database 5808.

The translation operation may translate or otherwise modify the format, language and/or data model of a semantic identifier. A translation operation may involve a translation or mapping to or from one or more data tools, languages, formats and/or data models to or from at least one other data tool, language, format and/or data model.

- 5 For example, a translation operation may involve a translation or mapping to, from or between known data integration tools, such as WebSphere DataStage 7 from IBM, WebSphere QualityStage from IBM, Business Objects tools, IBM - DB2 Cube Views, UML 1.1, UML 1.3, ERStudio, IBM's WebSphere ProfileStage, PowerDesigner (with added support for Packages and Extended Attributes) and/or MicroStrategy tools. A translation engine and/or translation operation may optionally be embodied in a metabroker. A translation
- 10 operation may be performed, executed and/or conducted in batch, real-time and/or on a continuous basis. A translation operation may be provided or made available as a service, for example, as part of a service oriented architecture. The SOA can be part of the infrastructure of an enterprise computing system of a business enterprise. In the SOA, services become building blocks for application development and deployment, allowing rapid application development and avoiding redundant code. Each service embodies a set of business logic or business
- 15 rules that can be blind to the surrounding environment, such as the source of the data inputs for the service or the targets for the data outputs of the service. As a result, services can be reused in connection with a variety of applications, provided that appropriate inputs and outputs are established between the service and the applications. The service-oriented architecture allows the service to be protected against environmental changes, so that the architecture functions even if the surrounding computer environment is changed. As a result, services may not need
- 20 to be recoded as a result of infrastructure changes, which may result in savings of time and effort. An SOA may be for a web service and may involve three entities, a service provider, a service requester and a service registry. The registry may be public or private. The service requester may search a registry for an appropriate service. Once an appropriate service is discovered, the service requester may receive code, such as Web Services Description Language ("WSDL") code, that is necessary to invoke the service. WSDL is a programming language
- 25 conventionally used to describe web services. The service requester may then interface with the service provider, such as through messages in appropriate formats (such as the Simple Object Access Protocol ("SOAP") format for web service messages), to invoke the service. The SOAP protocol is a preferred protocol for transferring data in web services. The SOAP protocol defines the exchange format for messages between a web services client and a web services server. The SOAP protocol uses an eXtensible Markup Language ("XML") schema, XML being a
- 30 generic language specification commonly used in web services for tagging data, although other markup languages may be used.

- Once a translation operation exists for a semantic identifier, database 112, database 112 including one or more semantic identifiers, system of information, system of information including one or more semantic identifiers or other item it can be translated to or from, mapped to, linked to, used with or associated with any other semantic
- 35 identifier, database 112, database 112 including one or more semantic identifiers, system of information, system of information including one or more semantic identifiers or other item sharing at least one translation operation. In embodiments, such as using an atomic data repository as a hub for a translation operation, the mapping of a translation operation can, among other things, trace data that is translated in the execution of the operation backward and forward between an original semantic context and a translated semantic context. Depending on the
- 40 context, the appropriate identifier for the data item may vary, such as by varying or truncating a syntax and/or string to enable more efficient storage or faster processing, or by varying the relationships used to form a unique identifier

where the semantic context varies. Thus, a dynamic identifier may combine the benefits of retraceable translation with the benefits of rapid processing, efficient data processing and effective operation in various contexts in which a data item is used.

A given item, such as an item that has an identity in a model, may exist in multiple forms or instances, such as a physical instance and a logical modeling instance. Figure 11 depicts an item, namely, a table of employee information 5902. However the concept or entity "employees" can exist in a number of different forms within an enterprise. For example, the employee table 5902 may exist as a physical table that stores values related to employees in a physical data storage facility. On the other hand, the entity employee may also be represented as a logical entity, such as an icon or text that represents employees in a logical modeling activity 5908, or in various other forms or instances. That is, the same item, including any associated data or metadata, may exist in multiple forms or instances across views, models, structures or a data integration environment, such as in databases, data repositories, models, hubs, or the like. Figure 12 depicts the employee table 5902 in one form or a single instance in a database 6002 and/or more than one form or instance in a database 6004 or hub 6008.

In order to distinguish between the various forms or instances of an item, any differentiating characteristic may be used, such as a level of abstraction, a physical property of an item, a location of the item within a hierarchy, a location of an item in a database, a context in which an item is found, a syntax of an item, a relationship of an item to other items, an attribute of an item, the class of an item, or other characteristic. For example, referring back to Figure 5, the items, or individuals in this case, may be distinguished based on age, gender, hair color, IQ, political affiliation and/or number of trips to the doctor in the past three months. For example, if age was selected as the product differentiator, it may be the case that Jessica is the only individual under ten years old, Betty is the only individual between fifty-seven and sixty-seven years old and Jim is the only individual who is thirty-seven years old. In another example, different forms or instances of the item may exist at different levels of abstraction or in different contexts. For example, the employee table may exist in multiple forms or instances in the hub 6102, such as a physical employee table 5904, such as used to store values in a database that relate to data that pertains to employees, and a logical employee model 5908, such as to be used in a view of process that relates to employees..

Distinguishing between the different instances of a particular identified item can enable a variety of other methods and processes. For example, in one embodiment, an item, such as a table named "employee," may be brought into a hub. A hub collector may have two forms or instances of "employee" in the hub; one corresponding to the physical database instance and another corresponding to the logical modeling activity. A differentiating characteristic, such as a property of the item attributed to the item in the hub allows for the differentiation between the physical instances and the logical model instances or forms. In embodiments that differentiating characteristic can be called a level of abstraction, such as to distinguish between logical and physical levels of abstraction. In other cases the hub may associate other characteristics with items, such as different forms of identifiers, relationships, classes, attributes, physical locations, logical positions, models and the like.

As depicted in Figure 14, when performing an operation, such as selecting data to be loaded into a database, translating data, generating a query, or the like, a system, such as a translation engine 6204, may grab, load or obtain all of the items from a hub 6208 or database 6210. It may select or filter 6204 the items based on any differentiating characteristic. For example, it may select or filter out those instances or forms that have a physical level of abstraction, that have a particular relationship to other items, that have a logical level of abstraction, that are created prior to a specified date and time, or that have any other distinguishing characteristics.. Thus, the methods

and systems described herein provide for selective handling of instances of the same item or entity based on any differentiating characteristic.

As depicted in Figure 15, when performing a data integration operation, such as a translation operation, which may be in response to a query 6202, a translation engine 6204 may filter or select items, including any data and/or metadata, at the hub 6208 or database 6210 and grab, load or obtain only those items of the relevant level of abstraction. For example, it may filter or select out those instances or forms with a logical level of abstraction, keeping only those with a physical level of abstraction. The filtering or selection may be performed at runtime or design time and may be conducted in batch, real-time or on a continuous basis. In embodiments such a method of filtering or selection may be provided as an RTI service in a services oriented architecture.

The filtering or selection may be based on information, such as a mapping of a data model, a mapping of a metadata model, a differentiating characteristic, a relationship of an item to another item, an attribute of an item, or the syntax of an identifier, that is obtained by the translation engine and/or system at development-time, design-time or run-time. In embodiments the information may be updated in a dynamic fashion in real-time.

The closer in the overall process the filtering or selection is to the hub or database the more efficient and faster the operation. As depicted in Figure 16, the translation engine 6204 may perform a translation operation on the query 6202 itself, resulting in a revised query 6402, which may be sent for further processing, such as directly to the hub 6208 or database 6210. For example, the revised query 6402 may be rendered in a format that is directly compatible with the native format of the hub 6208 or database 6210. For example, by rendering the query in the native format of the database 6210, the system may increase processing efficiency for the query. Similarly, the query 6402 may be filtered or a command such as a select command may be generated to keep a logical modeling entity rather than a physical entity, in which case the query 6402 may be rendered in a format suitable for a logical modeling activity (such as a graphical user interface), rather than for the database. Of course, not only queries but other messages and operations may be filtered according to level of abstraction, enabling the same entity to be tracked across the data integration platform and handled according to the suitable operating environment of a particular data integration activity.

The methods and systems described herein can be used to capture semantic contexts and to handle data integration tasks with respect to a wide range of items related to an enterprise, such as an object, data item, datum, column, row, table, database, instance, attribute, metadata, concept, topic, subject, semantic identifier, other identifier, RFID tag, vendor, supplier, customer, person, team, organization, user, network, system, device, family, store, product, product line, product feature, product specification, product attribute, price, cost, bill of materials, shipping data, tax data, course, educational program, location, map, division, organization, organism, process, rule, law, rating system, good, service and/or service offering.

The methods and systems described herein can be used in a variety of semantic contexts, such as a step in an enterprise method, a datum in a database, a datum in a row or column, a row or column in a table, a row or column in a database, a datum in a table, a table in a database, metadata in a database, an item in a hub or repository, an item in a database, an item in a table, an item in a column, an item in a row, a person in an organization, a sender or recipient of a communication, a user on a network, a system on a network, a device on a network, a person in a family, an item in a store, a dish on a menu, a product in a product line, a product in a product offering, a course or step in an educational or training program, a location on a map, a location of an item, a division of an organization, a person on a team, a rule in a system of rules, a service in a service suite, an entity in an organizational hierarchy of an enterprise, an entity in a supply chain, a customer in a market, purchaser in a

purchasing decision, a price of a good or service, a cost of a good or service, a component of a product or system, a step of a method, a member of a group, or many others.

While the invention has been described in connection with certain preferred embodiments, it should be understood that other embodiments would be recognized by one of ordinary skill in the art, and are intended to fall
5 within the scope of this disclosure.

CLAIMS

What is claimed is:

1. A method of data integration, comprising:
 providing a semantic identifier for identifying an item based on its relationship with another item;
 5 obtaining a mapping of a data model to enable determination of the semantic identifier for an item in the data model; and
 associating the mapping with a data integration function, wherein the execution of the data integration function is based on at least one of the mapping and the semantic identifier.
2. A method of claim 1, wherein the item includes one or more of an object, a data item, a datum, a column, a row, a table, a database, an instance, an attribute, metadata, a concept, a topic, a subject, an identifier, a semantic
 10 identifier, an RFID tag, a vendor, a supplier, a customer, a person, a team, an organization, a user, a network, a system, a device, a family, a store, a product, a product line, a product feature, a product specification, a product attribute, a price, a cost, a bill of materials, shipping data, tax data, a course, an educational program, a location, a map, a division, an organization, an organism, a process, a rule, a law, a rating system, a good, a service and a
 15 service offering.
3. A method of claim 1, wherein the relationship involves the position of an item in a relational hierarchy.
4. A method of claim 1, wherein the semantic identifier is a unique identifier for an item.
5. A method of claim 1, wherein the semantic identifier is based on less than all relationships of the item with other items, but a sufficient number of relationships to ensure the identifier is unique.
- 20 6. A method of claim 1, wherein the semantic identifier is based on the minimum number of relationships to ensure that the identifier is unique.
7. A method of claim 1, wherein the semantic identifier is a context-dependent identifier for an item.
8. A method of claim 1, wherein the semantic identifier is stored in atomic format.
9. A method of claim 1, wherein the semantic identifier is stored in atomic format in a data repository.
- 25 10. A method of claim 1, wherein the semantic identifier is dynamic.
11. A method of claim 1, wherein the semantic identifier varies by context.
12. A method of performing a data integration process, comprising:
 associating a model with a data set; and
 forming a select command to select items from the data set, wherein the form of the select command is
 30 based on a distinguishing characteristic for the items that is determined from the model.
13. A method of claim 12, wherein the formation of the select command/query is performed at runtime for a process that uses the select command/query.
14. A method of claim 12, wherein the formation of the select command/query is performed at the time of design of a process that uses the select command/query.
- 35 15. A method of performing a data integration process, comprising:
 associating a model with a data set; and
 forming a query to query the data set, wherein the form of the query is based on a distinguishing
 characteristic for the items that is determined from the model.
16. A method of claim 15, wherein the formation of the select command/query is performed at runtime for a
 40 process that uses the select command/query.

17. A method of claim 15, wherein the formation of the select command/query is performed at the time of design of a process that uses the select command/query.
18. A system for data integration, comprising:
a semantic identifier for identifying an item based on its relationship with another item;
5 a mapping of a data model to enable determination of the semantic identifier for an item in the data model;
and
a facility for associating the mapping with a data integration function, wherein the execution of the data integration function is based on at least one of the mapping and the semantic identifier.
19. A system of claim 18, wherein the relationship involves the position of an item in a relational hierarchy.
- 10 20. A system of claim 18, wherein the semantic identifier is a unique identifier for an item.
21. A system of claim 18, wherein the semantic identifier is based on less than all relationships of the item with other items, but a sufficient number of relationships to ensure the identifier is unique.
22. A system of claim 18, wherein the semantic identifier is based on the minimum number of relationships to ensure that the identifier is unique.
- 15 23. A system of claim 18, wherein the semantic identifier is a context-dependent identifier for an item.
24. A system of claim 18, wherein the semantic identifier is stored in atomic format.
25. A system of claim 18, wherein the semantic identifier is stored in atomic format in a data repository.
26. A system of claim 18, wherein the semantic identifier is dynamic.
27. A system of claim 18, wherein the semantic identifier varies by context.
- 20 28. A system of claim 18, wherein the semantic identifier recursively captures an indirect relationship to a second item by capturing a direct relationship to a first item that has a direct relationship to the second item.
29. A system of claim 18, wherein the semantic identifier is captured in a string and wherein the string is truncated if all elements are not required for a unique identifier.
30. A system of claim 18, wherein the data integration function is a translation operation.
- 25 31. A system of claim 30 wherein the translation operation modifies one or more of the format of a semantic identifier, the language of a semantic identifier, and the data model of a semantic identifier.
32. A system of claim 30 wherein the mapping of the translation operation can trace data that is translated in the execution of the operation backward and forward between an original semantic context and a translated semantic context.
- 30 33. A system of claim 30 wherein translation operation is provided as a service in a services oriented architecture.
34. A system of claim 18, further comprising a filter for selectively filtering instances of a logical entity based on a distinguishing characteristic of the entity.
35. A system of claim 34, wherein the distinguishing characteristic is derived from at least one of the mapping
35 and the semantic identifier.

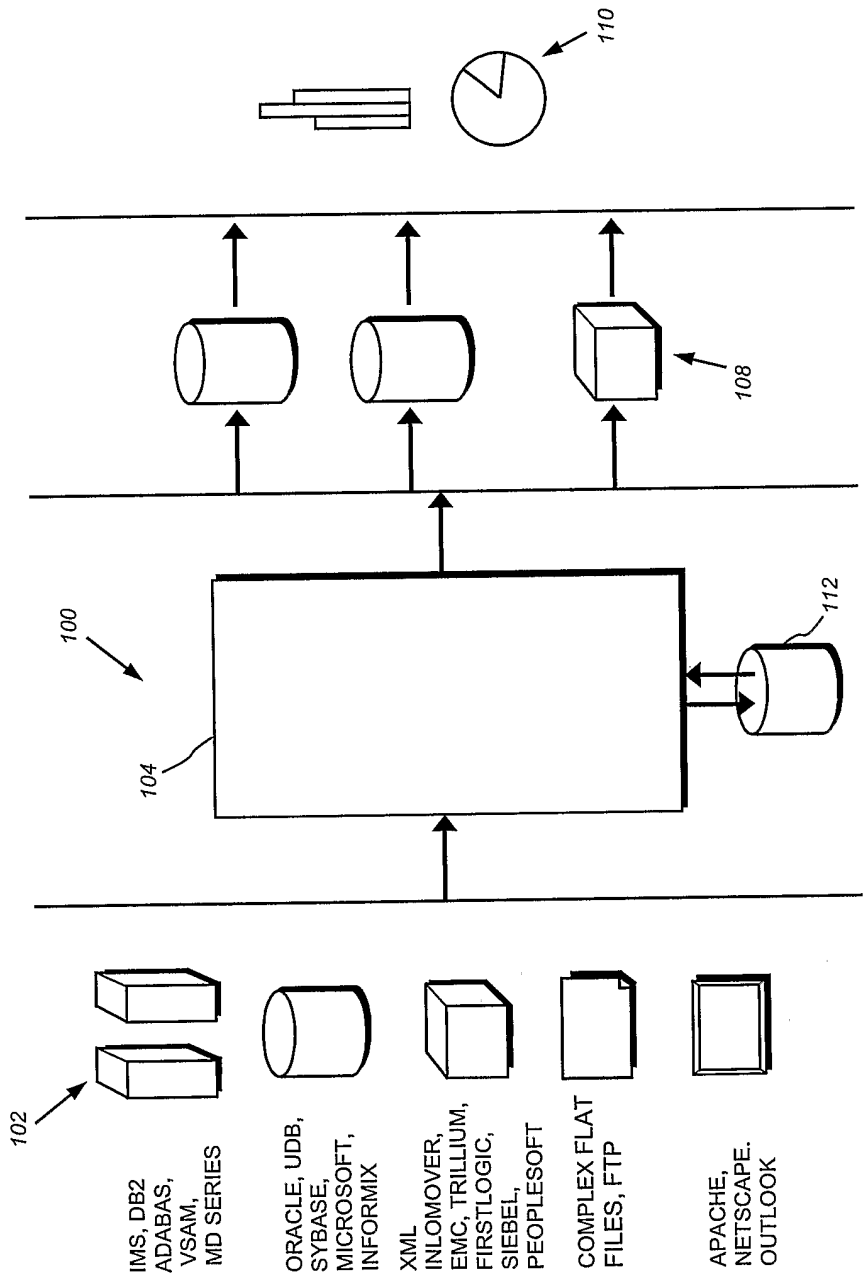
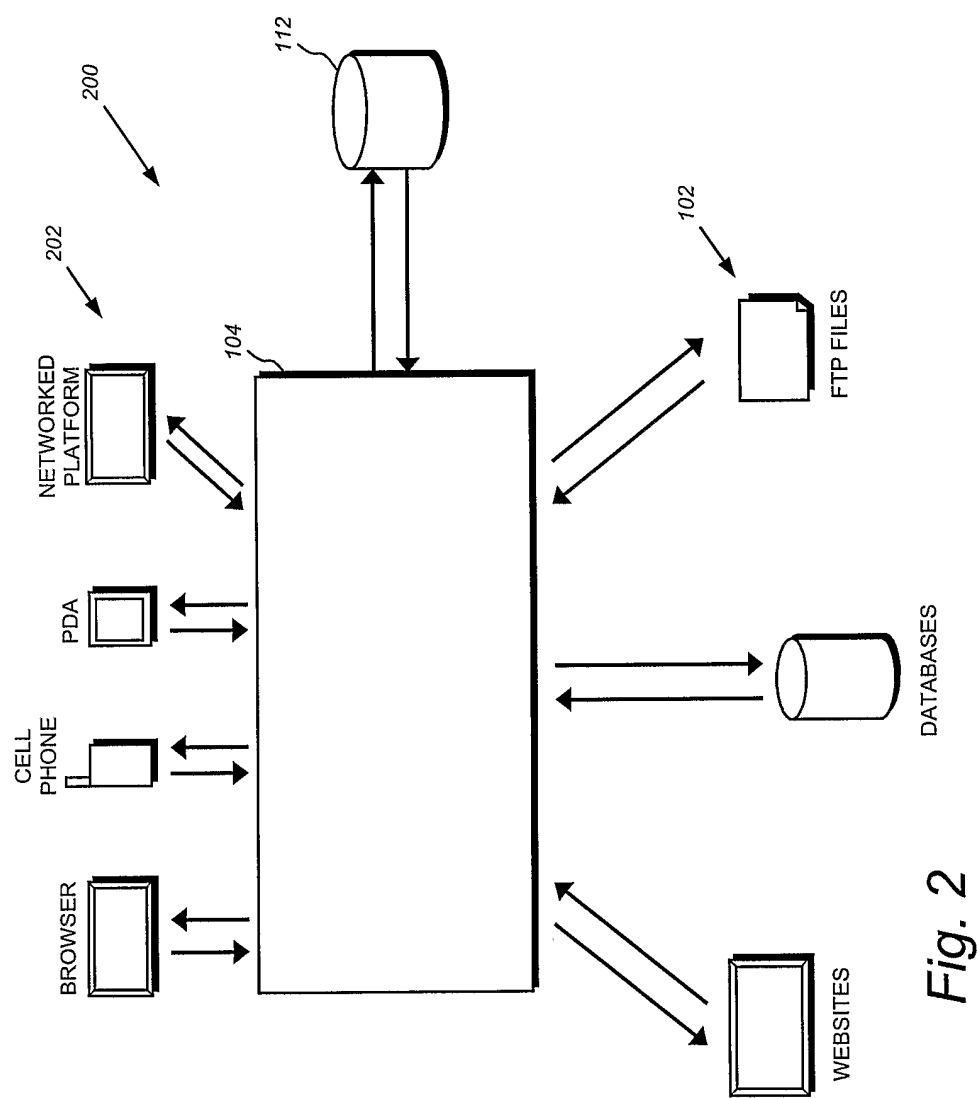


Fig. 1



3/17

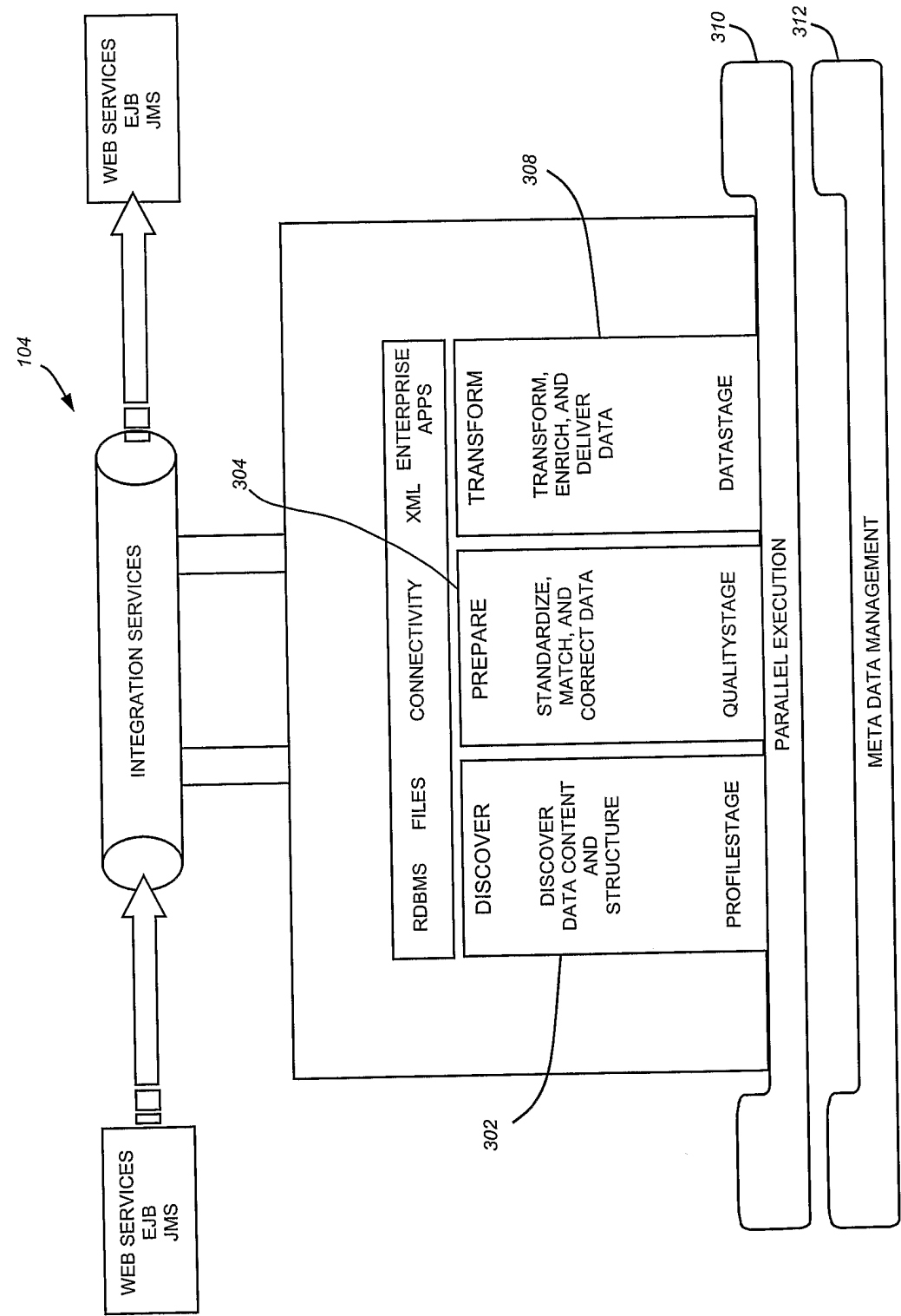


Fig. 3

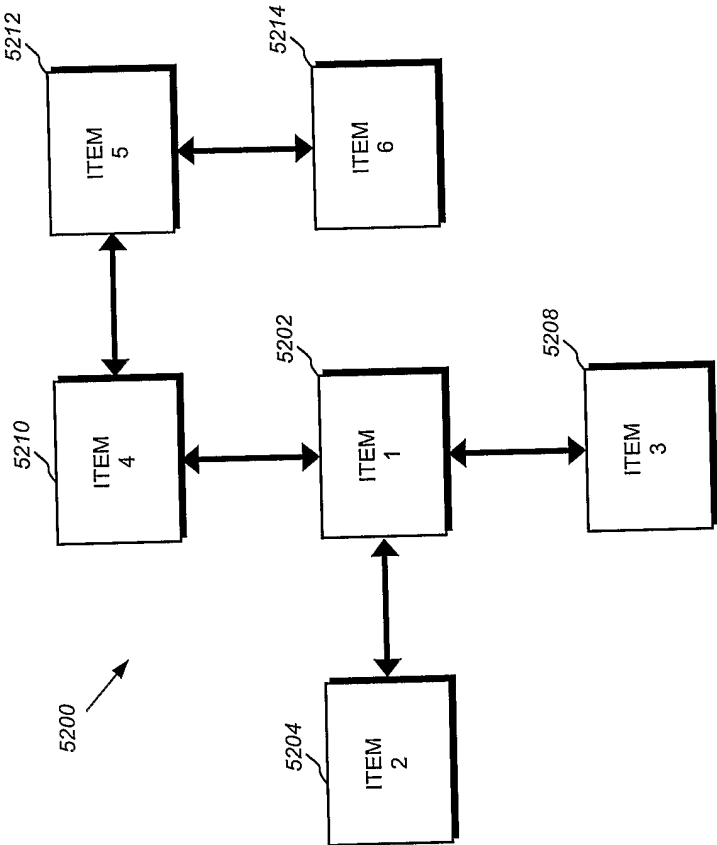
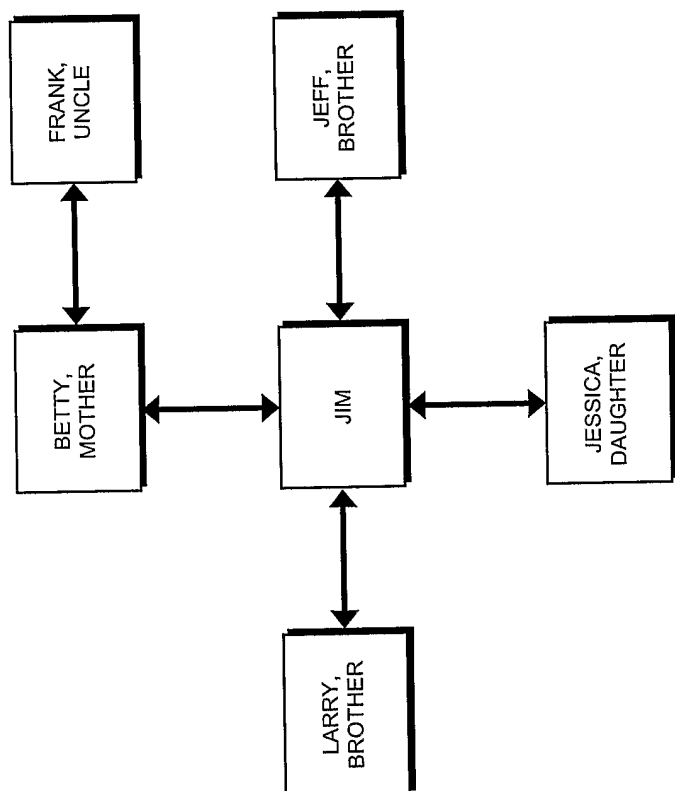


Fig. 4

5/17

*Fig. 5*

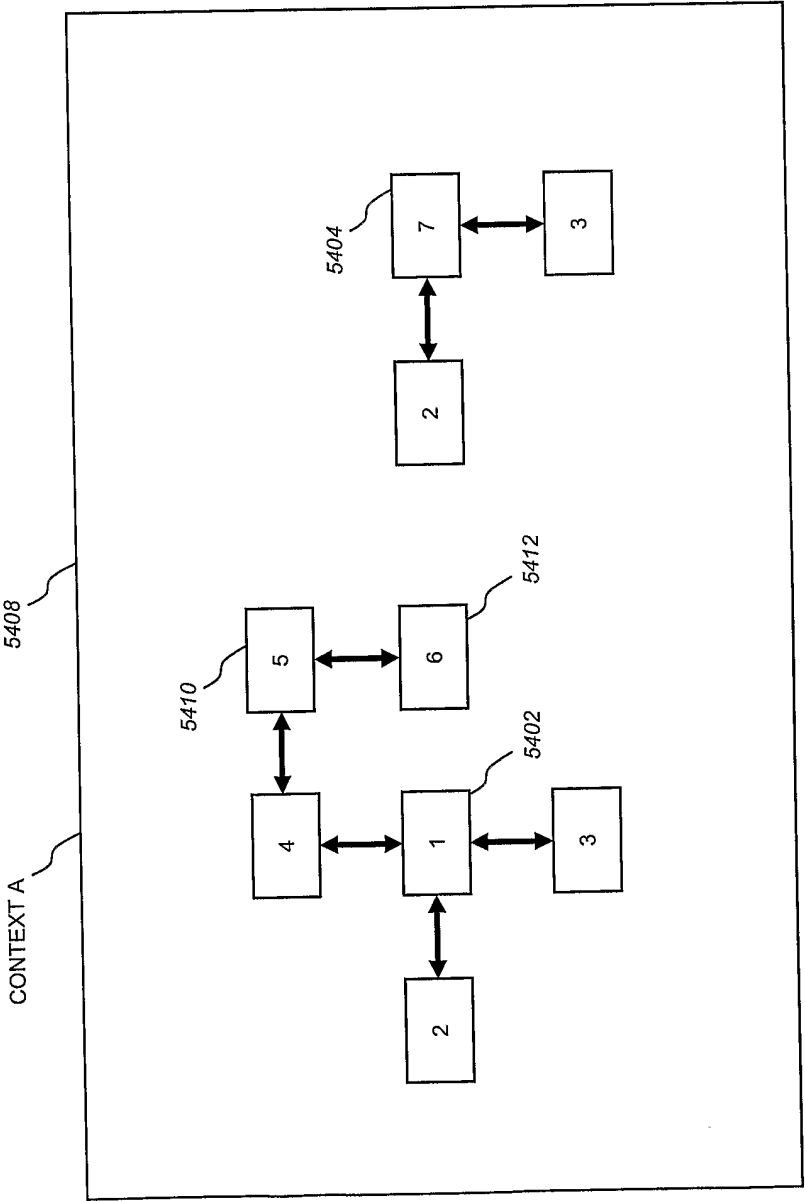


Fig. 6A

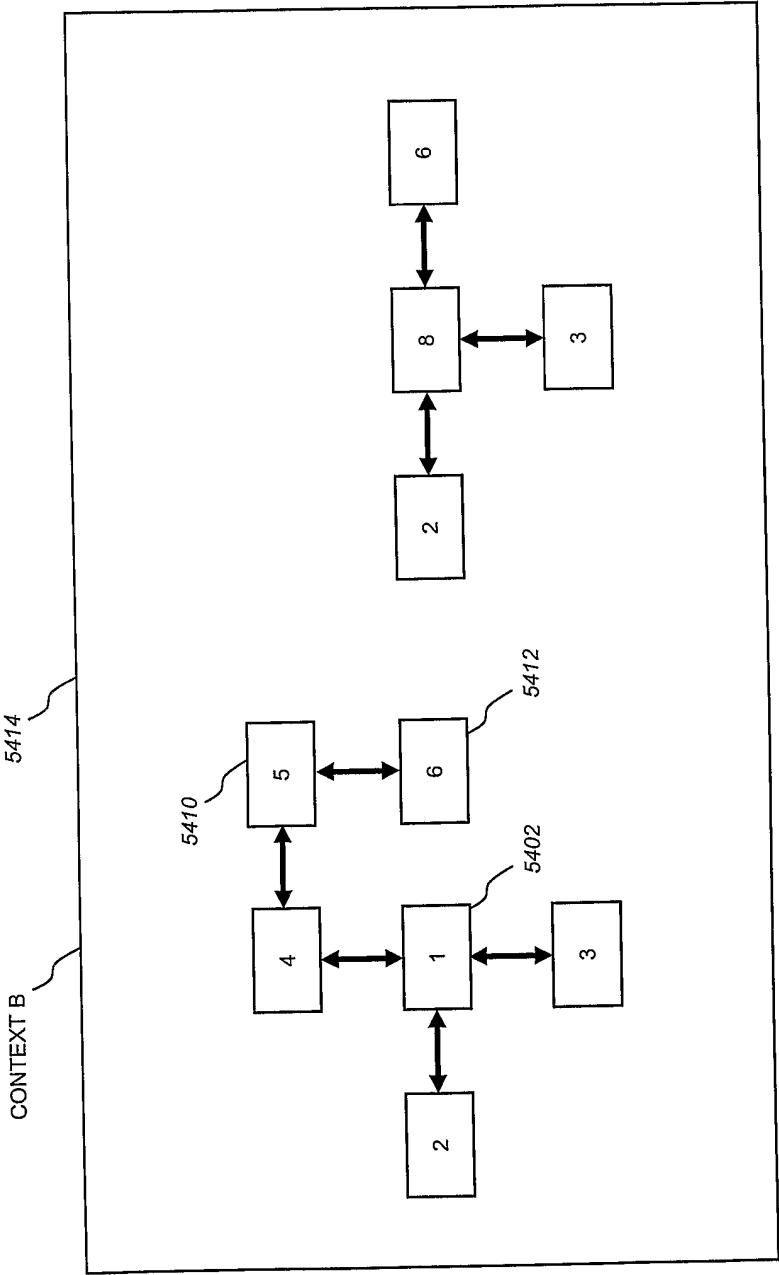


Fig. 6B

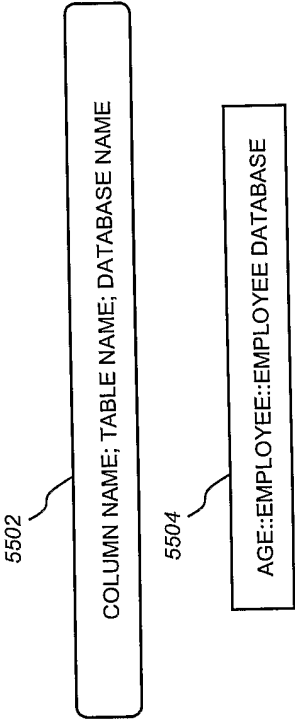


Fig. 7

9/17

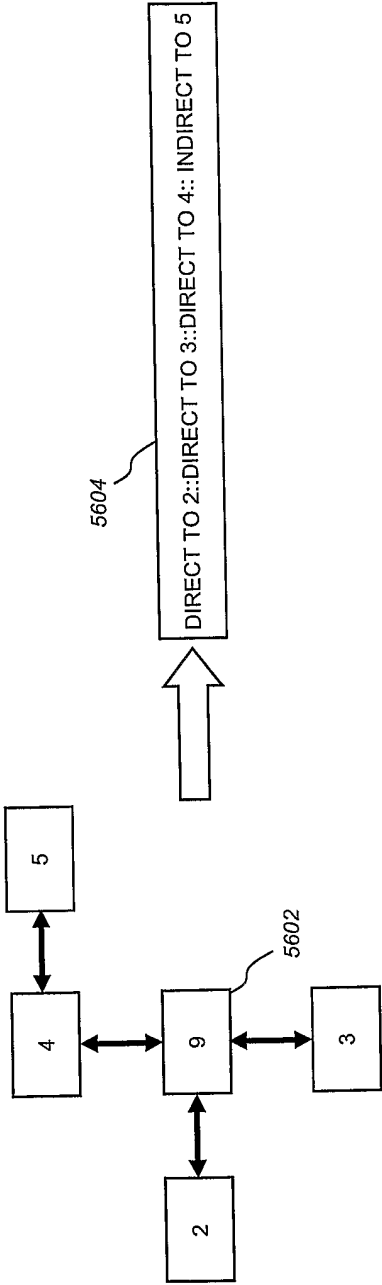


Fig. 8

10/17

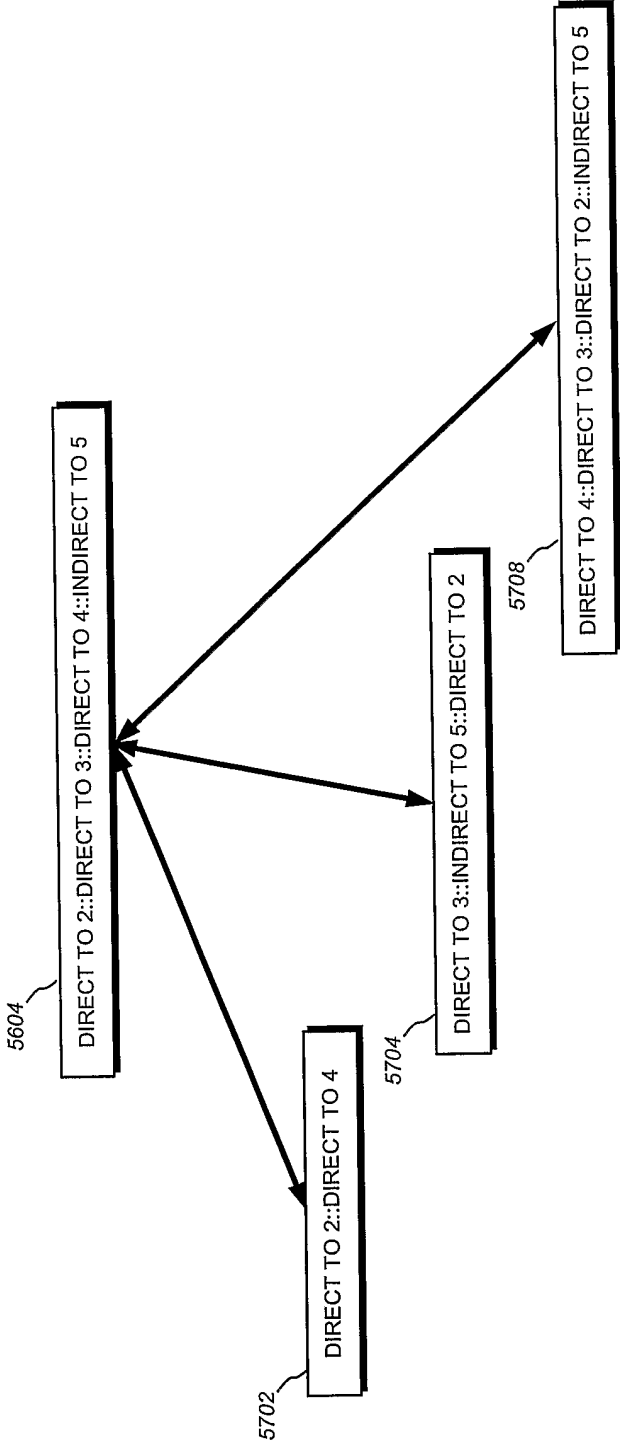


Fig. 9

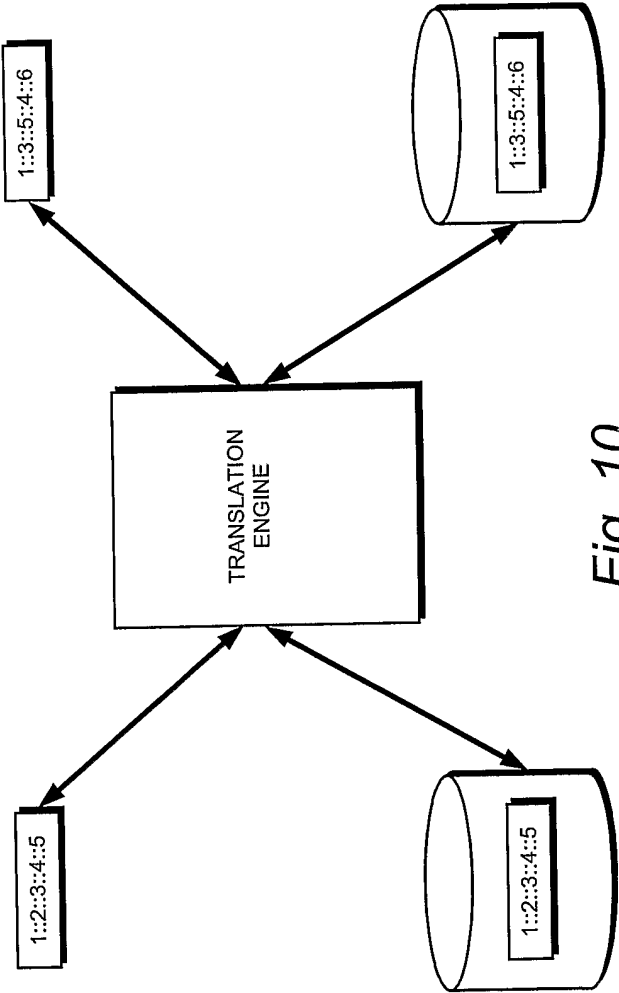


Fig. 10

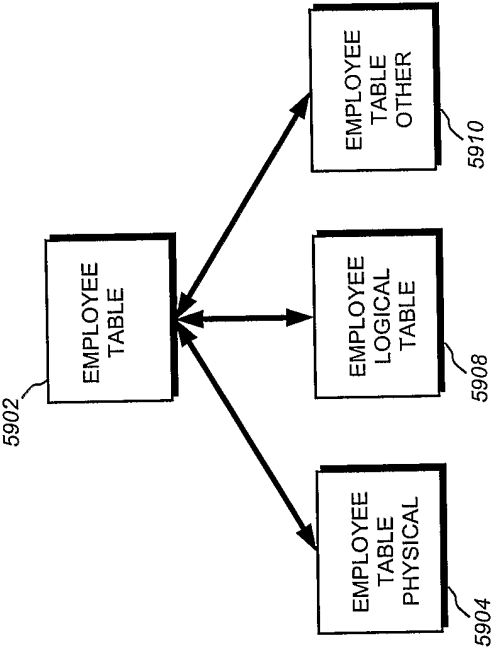


Fig. 11

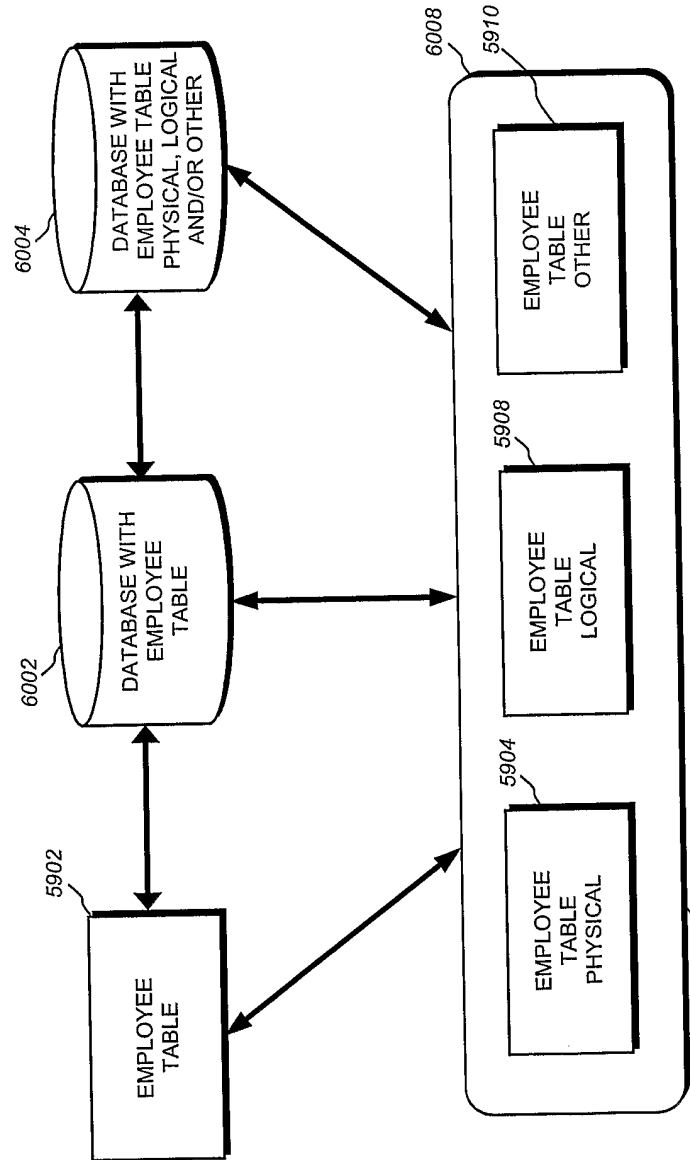


Fig. 12

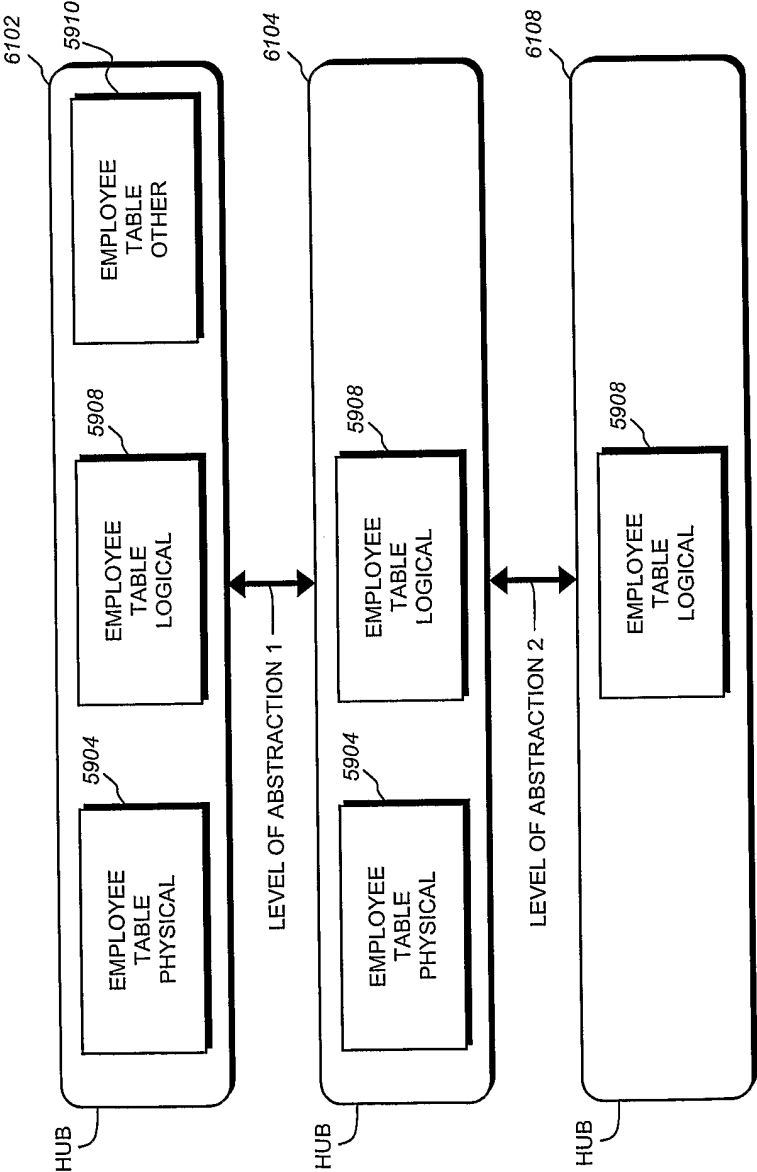


Fig. 13

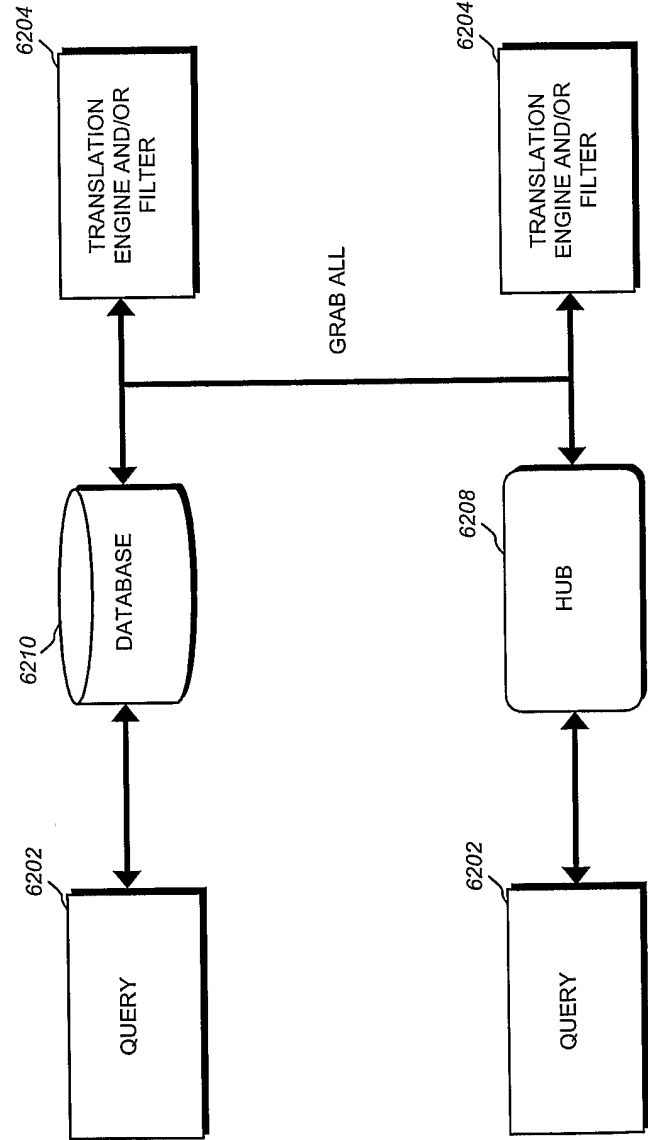


Fig. 14

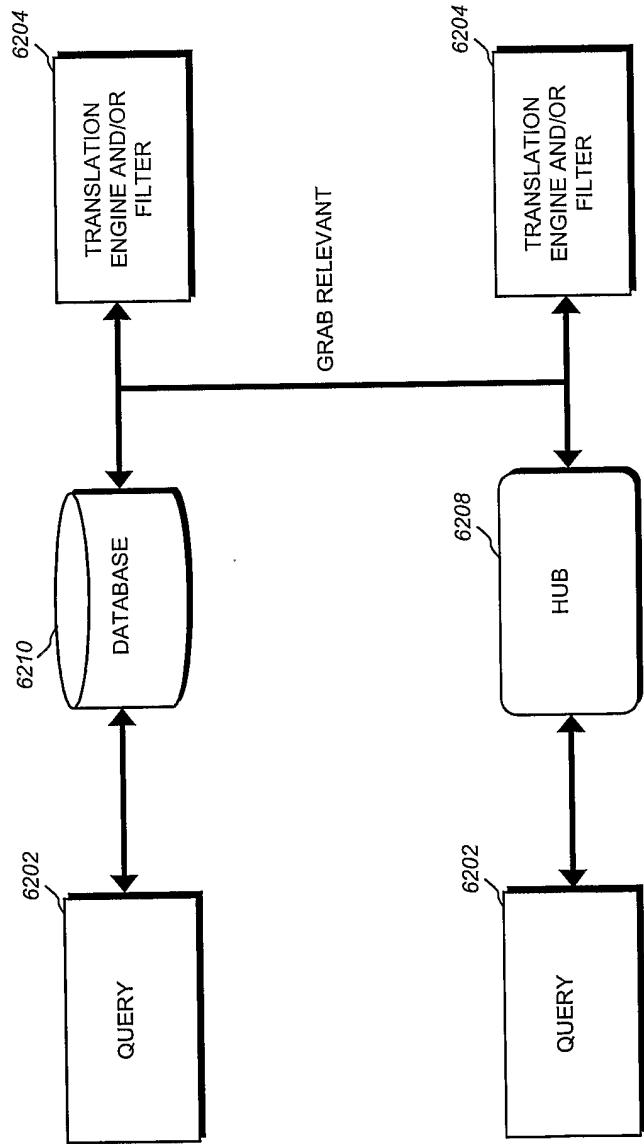


Fig. 15

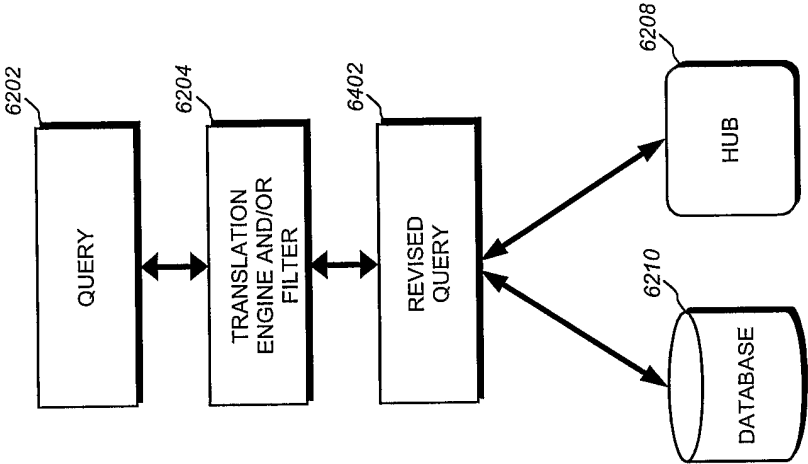


Fig. 16