(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 November 2013 (28.11.2013)

(10) International Publication Number
WO 2013/175422 A4

(51) International Patent Classification:
G06F 17/00 (2006.01)

(21) International Application Number:
PCT/IB2013/054254

(22) International Filing Date:
23 May 2013 (23.05.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/650,738 23 May 2012 (23.05.2012) US
PCT/IB2013/054191 22 May 2013 (22.05.2013) IB

(71) Applicant: **BI-BUILDERS AS** [NO/NO]; Koppholen 19, N-4313 Sandnes (NO).

(71) Applicant *(for MW only)*: **OPPEDAHL PATENT LAW FIRM LLC** [US/US]; P O Box 5940, Dillon, Colorado 80435 (US).

(72) Inventor: **FRAFJORD, Erik**; c/o Bi-Builders AS, Koppholen 19, N-4313 Sandnes (NO).

(74) Agent: **OPPEDAHL, Carl**; Oppedahl Patent Law Firm LLC, P. O. Box 5940, Dillon, CO 80435 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report (Art. 21(3))*
— *with amended claims (Art. 19(1))*

Date of publication of the amended claims: 13 February 2014

(54) Title: METHODOLOGY SUPPORTED BUSINESS INTELLIGENCE (BI) SOFTWARE AND SYSTEM

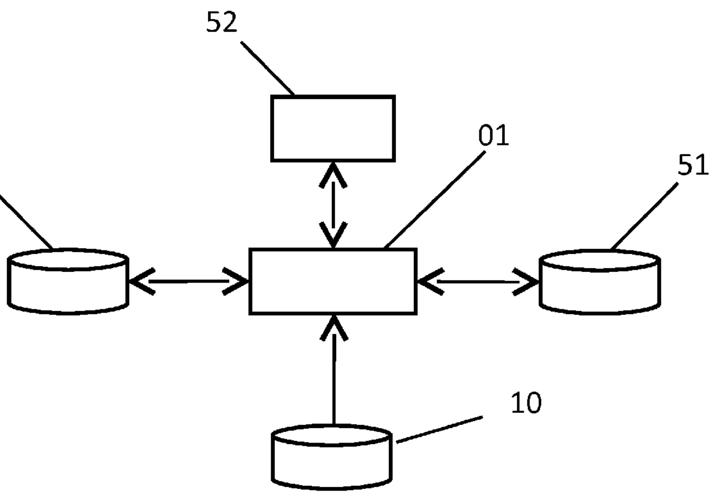

Fig. 7

(57) **Abstract**: The disclosed device provides idealized and reusable data source interfaces. The process of idealizing includes reengineering of an original data model using a surrogate key based model. The technique emphasizes readability and performance of the resulting operational data store. In, addition, the disclosed device provides a unique method for handling changes which allows for all types of changes to be automatically implemented in the operational data store by table conversion. Further the disclosed device provides inline materialization which supports a continuous data flow dependency chain. A continuous dependency chain is used to provide automated documentation as well as a dynamic paralleled transformation process.

AMENDED CLAIMS
received by the International Bureau on 29 December 2013 (29.12.2013)

1. A method of transforming raw electronic data which is stored in a first application data model into a second data model and loading data into said second data model, said method comprising the steps of:

(i) defining an idealized data model for at least one data source, said defining step comprising:

importing metadata from said data source into a product;

refining data model keys and relationships in said product if necessary; and

improving and/or selecting one or more new table- and column-names capable of defining said data source; and

(ii) processing an idealized data model for at least one data source by converting a first data model into a second relational data model, said processing step comprising:

converting one or more original data source keys and relationships to a surrogate key-based model by creating at least one destination table with an idealized name format;

creating a surrogate key conversion table for each destination table; and

importing data through a parallel processing of said destination tables.

2. The method of claim 1, wherein said importing of metadata further comprises an importing of tables, table names, column names, keys and relationships if information exists in a DBMS system.

3. The method of claim 1, wherein said importing of metadata further comprises an importing of table and column descriptions, key and relationship definitions if information exists in a data source application repository.

4. The method of claim 1, wherein said refining of data model keys and relationships further comprises an exporting of said data model to an empty metadata database, maintaining said data

model definition using one or more standard DBMS features in creating a refined metadata model, importing said refined metadata model into a product again, and wherein each of said refining steps is capable of being performed as an iterative process and at any time.

5. The method of claim 1, wherein said improving and/or selecting of new table- and column-names further comprises

an editing of names directly in the product or exporting table and column definitions into an external standardized format,

maintaining said table and column names in said external standardized format,

importing said definitions into product again, and wherein each of said improving and/or selecting steps is capable of being performed as an iterative process and at any time.

6. The method of claim 1, wherein said creating of said at least one destination table further comprises selecting an idealized table name that is prefixed by a data source name, version number and instance number automatically defined in said product, and one or more idealized column names.

7. The method of claim 6, wherein said one or more idealized column names further comprises a primary key column which is a surrogate key column inheriting its name from said idealized table name and comprising a data type integer, a foreign key column which is a foreign surrogate key column inheriting its name from said related table name and comprising a data type integer, and if more than one reference is associated with the same table, a suffix comprising original foreign key column(s) having said defined idealized column name.

8. The method of claim 1, wherein said creating of a surrogate key conversion table for each data table further comprises idealizing a table name with a defined naming structure to separate it from each of said data tables and selecting an idealized column name having a surrogate key column name inheriting its name from said idealized table name and which comprises a integer and an original key column inheriting its name from said defined idealized column name and which comprises a data type from said data source.

9. The method of claim 1, wherein said importing of data through a parallel processing further comprises

dividing a data stream into an insert- and/or an update- stream,

executing data loads in a logical order as derived from said data model relationships, and

creating and/or updating surrogate key tables during a load process, each of said data tables dependent on a successful processing of its surrogate key tables and/or its tables that are referenced as a foreign key.

10. The method of claim 1, wherein said defining step further comprises the establishing of import filters and selection criteria from one or more or none of incremental rules for table load with or without overlapping records, one or more column level selection criteria for each table, global data source selection criteria and column level functions to manipulate data on row level.

11. A method of ensuring consistency between a configured product repository and a destination operational data store when changes to one or more configurations occurs, said method comprising the steps of:

creating and maintaining a static reference model further comprising a storing of object information in one or more object extended properties in said operational data store;

on a table level, at least one extended property containing a data source table;

on a column level, at least one extended property per column created using a primary surrogate key having a static standardized value, a foreign surrogate key having a value of a corresponding external foreign key name, and ordinary columns having a corresponding data source column name; and

comparing one or more repository configurations and definitions with one or more extended properties in said static reference model.

12. The method of claim 11, wherein said comparing of one or more repository configurations and

definitions further comprises:

extracting definitions from said repository and producing a first intermediate internal table,

extracting definitions from said operational data store and producing a second intermediate internal table,

comparing said first and said second intermediate internal tables, creating a discrepancy script if inconsistencies are found, and displaying said discrepancies to a user along with a repair script that optionally can be executed.

13. A method of constructing an unbroken dependency chain for all data transformation tasks in a data warehouse, information management and/or business intelligence (hereinafter "a solution") environment, said method comprising the steps of:

(i) establishing a naming format for database objects comprising one or more tables or views for a data transformation process, each of said tables or views includable in said unbroken dependency chain via naming and format standardization which can be specified in a product;

(ii) standardizing said solution environment by incorporating at least three standardized databases, a first database holding an idealized data source, a second database holding one or more transformation processes, a third database holding a multidimensional star diagram structure to be accessed by an end user visualization application;

(iii) creating said unbroken dependency chain by structuring and storing information in said standardized databases, wherein one or more physical dependencies are extracted from at least one DBMS system table into a dependency structure within said product, one or more dependencies that are derived from said standardized naming convention promoted by said product includable in said dependency structure within said product, said product enabling a defining of logical dependencies or relationships in said product and storage of said dependency structure within said product; and

(iv) defining and scheduling flexible update processes in said product by

using a dynamic unbroken dependency chain by defining logical dependencies on one or more top level objects within said multidimensional structure,

defining processing groups by using one or more fact table objects as input,

dynamically creating and maintaining a complete list of objects to be automatically included in an update process via said dependency structure, and

loading data by parallel processing of all objects on the same level in said dependency structure to automatically maximize performance efficiency.

14. The method of claim 13, wherein said step of establishing a naming format for database objects further comprises a deriving of a destination table name from said view name, a specifying a primary key column and an optional surrogate key column through said product or by a standardized format in database view, and an optional loading of full data or incremental data through said product or by a standardized format in database view.

15. The method of claim 13, wherein said database objects in said name establishing step further comprise one or more stored procedures, said one or more stored procedures having a view format comprising a destination table name and an associated view parameter, said one or more stored procedures being dynamically referable to said destination table and said associated view parameter.

16. The method of claim 13, wherein said one or more stored procedures is capable of being automatically loaded into said one or more tables.

17. A method to transform raw electronic data into meaningful and useful information, comprising:

idealizing metadata from at least one data source into a relational model, comprising,

importing metadata into a repository connected to a product,

generating intuitive table and column names by mapping a friendly name to an original name by the product,

refining the metadata to include table keys and relationships even if this information may not be accessible in the data source;

importing data from the at least one data source to a staging data store for temporary storage;

importing table(s) primary key(s) from the staging data store to a surrogate data store creating a surrogate key table, wherein the surrogate data store converts all original keys and foreign key references to surrogate keys, an original key being mapped to a surrogate key, the surrogate key table reflecting the link between the original and surrogate keys, wherein during insert and update operations the surrogate key tables are used to create and maintain surrogate keys;

processing the table for extraction to an operational data store, wherein the table can successfully update the surrogate key table before processing, the table being updated during processing if a record with an actual surrogate primary key exists in the operational data store, the table being loaded if a record with the actual surrogate primary key does not exist in the operational data store;

importing data to said operational data store, wherein the table has to successfully update the corresponding surrogate key table and the surrogate key table(s) of any related tables before processing; and

performing a consistency check on metadata level by comparing the repository with the operational data store.

18. The method of claim 17, wherein the idealizing step further comprises exporting a metadata database to provide primary and foreign keys using standard DBMS functionality, and wherein a revised metadata database is imported back into said repository where it can be iteratively refined one or more times, the relational model being a reusable object that can be purchased as a commodity.

19. The method of claim 17, wherein the idealizing step further comprises establishing user-crafted user-selected table name mappings and user-crafted user-selected column name mappings which can be set forth in an external spreadsheet exported by the system, the system disposed to read the spreadsheet and to bring about the associations with respect to the tables in response to the content of the spreadsheet.

20. The method of claim 17, wherein the check performing step further comprises creating a first intermediate internal table extracting data from the repository, creating a second intermediate internal table extracting data from the operational data store, joining the first and second intermediate internal tables, creating a discrepancy script if any inconsistencies are found, exporting the operational data store table directly to a star schema database if discrepancies are not found, and exporting the operational data store table to a ETL data store to refine the table and export the table to the star schema database if discrepancies are found.

21. A system for transforming raw electronic data which is stored in a first application data model into a second data model and loading data into said second data model, said system comprising:

an idealized data model for at least one data source, said idealized data model comprising imported metadata from said data source, refined data model keys and relationships, and one or more new table- and column names capable of defining said data source;

said idealized data model for at least one data source capable of converting a first data model into a second relational data model;

one or more original data source keys and relationships convertable to a surrogate key-based model through the creation of at least one destination table with an idealized name format;

at least one surrogate key conversion table for each destination table; and

data imported through a parallel processing of said destination tables.

22. The system of claim 21 further comprising an empty metadata database, capable of receiving exported data from said data model and maintaining a data model definition, a refined metadata model created from one or more standard DBMS features which can be imported into a system product, said refined metadata model capable of being generated iteratively.

23. The system of claim 21, wherein said at least one destination table further comprises an idealized table name that is prefixed by a data source name, version number and instance number automatically defined in a system product, and one or more idealized column names.

24. The system of claim 23, wherein said one or more idealized column names further comprises a primary key column which is a surrogate key column inheriting its name from said idealized table name and comprising a data type integer, a foreign key column which is a foreign surrogate key column inheriting its name from said related table name and comprising a data type integer, and if more than one reference is associated with the same table, a suffix comprising original foreign key column(s) having said defined idealized column name.

25. The system of claim 21, wherein said surrogate key conversion table for each data table further comprises a table name with a defined naming structure to separate it from each of said data tables and an idealized column name having a surrogate key column name inheriting its name from said idealized table name and which comprises a integer and an original key column inheriting its name from said defined idealized column name and which comprises a data type from said data source.

26. The system of claim 21 further comprising a data stream capable of being divided into an insert- and/or an update- stream and one or more data loads executable in a logical order as derived from said data model relationships.

27. The system of claim 21 further comprising import filters and selection criteria from one or more or none of incremental rules for table load with or without overlapping records, or from one or more column level selection criteria for each table, or from global data source selection criteria and column level functions to manipulate data on a row level.