extracted data has been correlated between the various data sources. Additionally, data from dispersed storage locations may be inadvertently left out of the information provided to decision makers. For at least the above stated reasons, the large amounts of data that are frequently retained by businesses would become difficult or impossible to manage using a manual process, thus necessitating an automated correlation process. Thus, what is needed is an automated method of and system for correlating data held in various repositories throughout an organization.

## SUMMARY

[0004] The general inventive concepts encompass methods of and systems for correlating data that is stored in a plurality of data repositories throughout an organization.

[0004a] According to a first aspect of the present invention, there is provided a method of correlating data stored at a plurality of storage locations, the method comprising:

extracting first data representing an item of information from a first storage location;

extracting second data representing the item of information from a second storage location; and

using the first data and the second data to assess the validity of the second data by confirming that a data size of the second data is the same as a data size of the first data,

wherein, when the second data is determined to be invalid:

maintaining the first data; and

discarding the second data; and

wherein, when the second data is determined to be valid:

converting the second data into third data having a standardized data format; and

storing the third data as new data, and

wherein confirming that the data size of the second data is the same as the data size of the first data comprises:

receiving a count of reference objects in the second data;

calculating an expected number of reference objects by determining a number of reference objects in the first data; and

comparing the received count to the calculated number and determining a variance between the received count and the calculated number.

[0004b] According to a second aspect of the present invention, there is provided a system for correlating data from a plurality of data sources, the system comprising:

a processor;

a hardware data interface;

a correlation engine database;

a memory;

wherein the processor is in electronic communication with the hardware data interface, the correlation engine database, and the memory;

the memory comprising software instructions that, when executed by the processor, cause the processor to:

extract, using the hardware data interface, first data representing an item of information from a first storage location;

extract, using the hardware data interface, second data representing the item of information from a second storage location;

receive, using the hardware data interface, information relating to the second data comprising:

file record quantity;

data source port data; and

a value representing a quantity of reference objects in the second data;

assess the validity of the second data using the first data by confirming that a data size of the second data is the same as a data size of the first data;

wherein, when the second data is determined to be invalid:

maintaining the first data; and

discarding the second data; and

wherein, when the second data is determined to be valid:

converting the second data into third data having a standardized data format; and

storing, using the hardware data interface, the third data as new data, and

wherein confirming that the data size of the second data is the same as the data size of the first data comprises:

receiving a count of reference objects in the second data;

calculating an expected number of reference objects by determining a number of reference objects in the first data; and

comparing the received count to the calculated number and determining a variance between the received count and the calculated number.

[0004c] According to a third aspect of the present invention, there is provided a non-transitory computer readable medium configured to store instructions for correlating data stored at a plurality of storage locations, wherein when executed, the instructions cause a processor to perform the following:

extracting first data representing an item of information from a first storage location;

extracting second data representing the item of information from a second storage location; and

using the first data and the second data to assess the validity of the second data by confirming that a data size of the second data is the same as a data size of the first data,

wherein, when the second data is determined to be invalid, the instructions, when executed, cause the processor to further perform the following:

maintaining the first data; and

discarding the second data; and

wherein, when the second data is determined to be valid, the instructions, when executed, cause the processor to further perform the following:

converting the second data into third data having a standardized data format;

storing the third data as new data in a third storage location; and

provide an interface that permits direct access to the third data, and

wherein confirming that the data size of the second data is the same as the data size of the first data comprises:

receiving a count of reference objects in the second data;

calculating an expected number of reference objects by determining a number of reference objects in the first data; and

comparing the received count to the calculated number and determining a variance between the received count and the calculated number.

[0005] In an exemplary embodiment, a method of correlating data among the various storage locations and data sources involves providing the data to a correlation engine. The correlation engine extracts data from each of these storage locations and data sources and analyzes each piece of data in order to determine if the received data is current with regard to other copies of the data stored across the organization.

[0006] In some exemplary embodiments, the extraction of data is performed using a collection script.

[0007] In some exemplary embodiments, the correlation engine extracts data from a storage location and compares the extracted data to other copies of the data that were extracted from the same storage location.

[0008] In some exemplary embodiments of the invention, the correlation engine extracts data from a storage location and compares the extracted data to other copies of the data that were created before the extracted data.

[0009] In an exemplary embodiment, a system for correlating data among the various storage locations and data sources. The system comprises a plurality of data sources, storage locations, and a correlation engine. The correlation engine receives data from each of these storage locations and data sources and analyzes each piece of data in order to determine if the received data is current with regard to other copies of the data stored across the organization. In this manner, it is possible to ensure that the data is properly associated with, corresponding, or otherwise related data throughout the organization.

CLAIMS

1. A method of correlating data stored at a plurality of storage locations, the method comprising:

extracting first data representing an item of information from a first storage location;

extracting second data representing the item of information from a second storage location; and

using the first data and the second data to assess the validity of the second data by confirming that a data size of the second data is the same as a data size of the first data,

wherein, when the second data is determined to be invalid:

maintaining the first data; and

discarding the second data; and

wherein, when the second data is determined to be valid:

converting the second data into third data having a standardized data format; and

storing the third data as new data, and

wherein confirming that the data size of the second data is the same as the data size of the first data comprises:

receiving a count of reference objects in the second data;

calculating an expected number of reference objects by determining a number of reference objects in the first data; and

comparing the received count to the calculated number and determining a variance between the received count and the calculated number.

2. The method of claim 1, wherein using the first data and the second data to assess the validity of the second data comprises: confirming that there were no reported errors during a process of extracting the second data from the second storage location.

3. The method of claim 1, wherein using the first data and the second data to assess the validity of the second data further comprises:

parsing the first data to obtain first parse data corresponding to one or more parse categories;

parsing the second data to obtain second parse data corresponding to the one or more parse categories; and

comparing the first parse data and the second parse data.

4. The method of claim 1, further comprising:

providing an interface that permits direct access to the third data.

5. The method of claim 1, further comprising, if the second data is determined to be valid: logging the second data as historical data.

6. The method of claim 1, further comprising receiving information related to the first storage location wherein such information comprises:

port identifier data;

zoning data; and

storage location configuration data, and

wherein the information related to the first storage location is stored outside of the first storage location.

7. The method of claim 1, further comprising receiving information related to the first storage location wherein such information comprises:

fabric login data;

fabric channel name server data;

interface data; and

zoneset data.

8. The method of claim 1, further comprising: storing the third data in a third storage location.

9. The method of claim 8, wherein storing the third data in the third storage location further comprises:

moving a previously stored data value representing the item of information from the third storage location into a previous day storage table; and

storing the third data into the data storage location in the third storage location.

10. A system for correlating data from a plurality of data sources, the system comprising:

a processor;

a hardware data interface;

a correlation engine database;

a memory;

wherein the processor is in electronic communication with the hardware data interface, the correlation engine database, and the memory;

the memory comprising software instructions that, when executed by the processor, cause the processor to:

extract, using the hardware data interface, first data representing an item of information from a first storage location;

extract, using the hardware data interface, second data representing the item of information from a second storage location;

receive, using the hardware data interface, information relating to the second data comprising:

file record quantity;

data source port data; and

a value representing a quantity of reference objects in the second data;

assess the validity of the second data using the first data by confirming that a data size of the second data is the same as a data size of the first data;

wherein, when the second data is determined to be invalid:

maintaining the first data; and

discarding the second data; and

wherein, when the second data is determined to be valid:

converting the second data into third data having a standardized data format; and

storing, using the hardware data interface, the third data as new data, and

wherein confirming that the data size of the second data is the same as the data size of the first data comprises:

receiving a count of reference objects in the second data;

calculating an expected number of reference objects by determining a number of reference objects in the first data; and

comparing the received count to the calculated number and determining a variance between the received count and the calculated number.

11. The system of claim 10, wherein using the first data and the second data to assess the validity of the second data comprises: confirming that there were no reported errors during a process of extracting the second data from the second storage location.

12. The system of claim 10, wherein using the first data and the second data to assess the validity of the second data further comprises:

parsing the first data to obtain first parse data corresponding to one or more parse categories;

parsing the second data to obtain second parse data corresponding to the one or more parse categories; and comparing the first parse data and the second parse data.

13. The system of claim 10, wherein the third data is stored in a third storage location.

14. The system of claim 13, wherein the third data is stored in the third storage location by:

moving a previously stored data value representing the item of information from the third storage location into a previous day storage table; and

storing the third data into the data storage location in the third storage location.

15. The system of claim 10, further comprising software instructions that, when executed by the processor, cause the processor to:

provide an interface that permits direct access to the third data.

16. The system of claim 10, wherein the software instructions comprise instructions

which, when executed by the processor, cause the processor to receive information related to the first storage location comprising:

port identifier data;

zoning data; and

storage location configuration data, and

wherein the information related to the first storage location is stored outside of the first storage location.

17. The system of claim 10, wherein the software instructions comprise instructions which, when executed by the processor, cause the processor to receive information related to the first storage location comprising:

fabric login data;

fabric channel name server data;

interface data; and

zoneset data.

18. A non-transitory computer readable medium configured to store instructions for correlating data stored at a plurality of storage locations, wherein when executed, the instructions cause a processor to perform the following:

extracting first data representing an item of information from a first storage location;

extracting second data representing the item of information from a second storage location; and

using the first data and the second data to assess the validity of the second data by confirming that a data size of the second data is the same as a data size of the first data,

wherein, when the second data is determined to be invalid, the instructions, when executed, cause the processor to further perform the following:

maintaining the first data; and

discarding the second data; and

wherein, when the second data is determined to be valid, the instructions, when executed, cause the processor to further perform the following:

converting the second data into third data having a standardized data format;

storing the third data as new data in a third storage location; and

provide an interface that permits direct access to the third data, and

wherein confirming that the data size of the second data is the same as the data size of the first data comprises:

receiving a count of reference objects in the second data;

calculating an expected number of reference objects by determining a number of reference objects in the first data; and

comparing the received count to the calculated number and determining a variance between the received count and the calculated number.

19. The non-transitory computer readable medium according to claim 18, wherein storing the third data in the third storage location further comprises:

moving a previously stored data value representing the item of information from the third storage location into a previous day storage table; and

storing the third data into the data storage location in the third storage location.

20. The non-transitory computer readable medium according to claim 18, wherein using the first data and the second data to assess the validity of the second data further comprises:

parsing the first data to obtain first parse data corresponding to one or more parse categories;

parsing the second data to obtain second parse data corresponding to the one or more parse categories; and

comparing the first parse data and the second parse data.