"System and method for processing complex queries"

## TECHNICAL FIELD

The present invention relates in general to the field of data processing. More specifically it concerns the handling of complex queries which here means that each query involves to search for and to retrieve from a database a multiplicity of items to be associated together to present to the user sets of said items combined together in a logical way.

One non-limitative application is the processing of search requests in the field of the travel and tourism industry when a user wishes to book a bundle of services, such as a combination of air travel services with hotel accommodation and/or car rental which are very different components by nature.

## BACKGROUND

Today's search systems are well efficient for processing requests for one type of items. In the travel industry field, search engines are now widely used and return travel solutions are sent in reply to an end user request. This is the case for engines dedicated to the airline industry such as low fare search engines providing the user with a set of low price flight solutions to be displayed on a friendly graphical interface.

The success of these techniques led the industry to propose computerized tools for enlarging the scope of the searches, for returning packages of products. This is particularly the case in the travel industry where the customer often needs a bundle of services when traveling. Products to be packaged can include flight tickets, hotel nights, and tourism activities.

Figure 1 depicts how a state-of-the-art technique builds product packages. A plurality of data repositories Da, Db and Dc are used and each stores travel products of one type. By way of example, Da, Db and Dc respectively contain information about air seats, about hotel rooms and about car rental services. A server S is used as a central node of the system. It receives user requests and returns travel packages solutions from end users over a network such as a wide area network preferably the internet. Upon receipt of a request, the server S communicates a plurality of individual queries each targeting one search engine Ea, Eb, Ec respectively associated with one of the repositories Da, Db, Dc. Local and independent searches are performed at the level of the search engines Ea, Eb, Ec and products are returned to the server S. The ~~later~~ latter server then aggregates the products to construct travel packages made of a plurality of products.

Such a search system is complex and involves multiple hardware/ software computer resources.

## SUMMARY

The invention aims at overcoming at least some of the drawbacks of current techniques for processing complex queries.

One technical problem the invention intends to solve is to optimize the processing of complex queries where multiple items of different natures need to be searched and combined to produce a response to a user request. The invention provides an efficient solution for searches based on heterogeneous data and triggered by complex queries.

In a more specific manner, the present invention relates to a method for processing a data search based on a complex query for at least one set of combined elements matching search parameters, at least two of the combined elements being of different element types. Said method preferably comprises the query processing steps of:

i) obtaining definition data of elements of a least two different element types, said definition data having heterogeneous data structures, said definition data comprising at least one element fare for each element,

ii) converting said definition data into converted definition data sharing one single generic data structure,

iii) receiving the complex query at a search engine,

iv) at the search engine, finding elements of at least two different element types whose converted definition data matches the query and combining said elements, the step of combining comprising:

- associating element fares of said elements,
- building at least one fared set of combined elements based on association of element fares

v) and returning at least one fared set of combined elements.

The at least one set of claims combined elements is/are preferably the best set(s) with regard to evaluation parameters such as, but not systematically, the price of the set. In the ~~later~~ latter case, the best set(s) is (are) the cheapest one(s).

Thanks to the invention, searches can be performed on the basis of very heterogeneous data sources. Heterogeneous data structures here means that at least two of the element types are defined by data of different data structures. The presentation of the data and/or the data fields describing the elements of said element types ~~are~~ is thus different from one source of data to at least another and this means that no single search engine is normally capable of searching among these heterogeneous sources. Whereas current systems cope with this problem by multiplying the search engines (one for each source) and by implementing post-processing means to aggregate the results obtained from the plurality of engines, the invention dramatically saves computer resources since one engine is sufficient. In addition the engine directly produces element associations without requiring a post step of aggregation.

And the results returned by the engine are much more reliable. This is because the engine directly works with elements described by definition data of the same data structure so that the search and the combination of the elements are integrated within the same process.

Further optional features which may be used cumulatively or alternatively are introduced hereafter:

- the method comprises the steps of storing the definition data in a repository and of making the search engine access the repository.
- the step of combining comprises checking if the association of element fares complies with combinatory rules.
- the complex query includes data defining a plurality of components derived from the search parameters and for which at least one element is to be searched for.

for each of the associations of matching elements, if it complies with the combinatory rules, validating said association, and it fails to comply with the combinatory rules, invalidating said association,

storing the validated associations or at least some of them; selected according to evaluation criteria, such as the price, to store only the best validated associations,

upon receipt of a request of an end user, retrieving at least one of said best validated associations which matches the request,

returning to the end user the at least one of said best validated associations;

- the definition data comprises fare data, availability data and application rule data;
- the element types include at least two of: air travel element type, railway travel element type, accommodation element type, car rental element type;
- the method is performed as a result of execution of computer software by at least one data processor, the computer software being stored in a non-transitory computer-readable memory medium, which here comprises any computer readable memory such as ROM or RAM memories.

The present invention further relates to a system for processing a data search based on a complex query comprising means for executing the method of the invention.

According to a preferred embodiment, the search engine is a low fare search engine.

This system may comprise computer software stored in a non-transitory computer-readable memory medium that is executed by at least one data processor that comprises part of the system.

The present invention also relates to a computer program product storing a computer program comprising instructions adapted to perform the method of the invention.

## BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will now be described in details with reference to the drawings for the purpose of illustrating the preferred embodiment.

Figure 1 illustrates an example of a state-of-the-art system for aggregating results of queries.

Figure 2 shows a preferred embodiment of one aspect of the invention related to the conversion of definition data.

Figure 3 depicts an example of conversion of definition data.

- *OR*
- *Passenger must be a clergyman*

- *Category 12 – Surcharges*
  - *Record 2*
    - *Between Paris and London*
    - *String with 1 record 3*
      - *If passenger flights in Business class he will pay a surcharge of 2000$*

This shows the complexity of definition of the elements when they are made of air travel products defined with fares. In addition the data structure and thus the element types may be different from one airline carrier or travel provider to another. And other travel elements (such as railways tickets, car rental or hotel rooms) have different definition data structures so that one technical problem is to work with the various data structures within the same process of response to a query.

Figure 2 depicts a first part ~~parts~~ of the invention corresponding to a conversion stage. D1, D2 and D3 represent data repositories - the number of which is not limited – and containing definition data of elements. By way of example Air travel elements are stored in D1, Hotel room elements are stored in D2 and car rental elements are stored in D3. The definition data of the elements stored in D1, D2 and D3 have different data structures.

For example, Air travel elements may be of an element type defined by:

- fare data
- availability data
- rules
- TLA data, i.e. data coming from low-cost carriers
- OAG data, i.e. flight data

Hotel room elements of D2 source may be of an element type defined by:

- one or plural fare data
- hotel and hotel rooms definition data
- rules

And car rental elements of D3 may be of an element type defined by:

- one or plural fare data
- cars definition
- rules

For illustration purpose, consider a request ~~form~~ from an origin to New-York city with accommodation, then the input of the request would involve three components: air (outbound), hotel, air (inbound), corresponding to two element types.

Preferably the request is transmitted over a communication network to a server sideNext, the definition of the relevant element types and the search parameters are used to define a query to be processed by a search engine to search for matching elements. The engine may be a conventional search engine such as a fare shopping engine used in transactional environments to return priced products. More precisely the engine may be a low fare search engine aimed at returning low price solutions. Thanks to the invention only one engine – and of a conventional kind – is sufficient to process the query which is however complex since it involves plural element types.

To search for matching elements, the engine accesses converted definition data obtained according to the previously described conversion method.

The processing of the queries is preferably driven by the element fares of the elements. As previously indicated, each element is defined by general data (identification of the element; general definition data...) and by at least one (but often plural) element fares. The association is here based on the element fares. Upon receipt of a query, candidate elements are searched for each component of the query. For the found elements, the system starts an association mechanism in which element fares of the found elements are associated. The system discards associations that failed (when their rules are not compatible) and generates successful element fares associations. The ~~later~~ latter associations are then used to build sets of combined elements to which a fare is assigned deriving from the association of the element fares. Such sets of combined elements are thus called fared sets of combined elements.

It should be noted that one element can be the source of series of element fares associations. For example, one element having one element fare A and one element fare B ~~lead~~ leads to associations based on A and or B.

In one embodiment, the best elements found for each component are combined.

However other association policies may be pursued.

Associations of elements must comply with the combinatory rules. In a preferred embodiment, this check is done dynamically with the association process: starting from one matching element fare (corresponding to an element included in a set of elements matching one component of the request), a second matching element fare (corresponding to an element included in a set of elements matching another component of the request) is considered. Their combinability is checked by accessing the combinatory rules. This check is depicted on figure 5.

If the association is invalidated, it is withdrawn from consideration by the engine which continues its search.

If the association is validated, the engine then tries to combine the association with another matching element fare of an element corresponding to still another component of the request. Again combinatory rules are checked and the process continues until a complete association of matching elements fitting the whole request has ~~be~~ been determined and checked against the combinatory rules.

Returning to figure 4, the output of the process consists in at least one validated association of elements complying with the search parameters.

Another flow is illustrated in figure 6 for processing the user request.

According to this embodiment, associations of elements are built during a construction stage. They are stored for future use. This corresponds to a pre-process in which massive queries are generated at the server side without direct link to an actual user request.

The queries are processed by an engine which can be the same as the engine described in reference to figure 4. At end user transaction time, another engine will parse the database with all cheapest associations, and discard the ones invalid due to non-combinability.

The stored associations are accessed by the second search engine upon receipt of a user request. Matching associations are then confronted to the combinatory rules and non-complying associations are invalidated. Validated associations are returned to the user. From a user point of view, the response time is decreased thanks to this embodiment. A drawback is that all calculated associations may be invalidated by this post-filtering.

A variation of the embodiment of figure 6 is given in figure 7. Associations of elements are built in the same manner but they are immediately checked against the combinatory rules. This leads to validated associations then stored for future use. Upon receipt of a request, the search engine accesses the validated associations and tries to find associations matching the search parameters of the user request. Such matching validated associations can be returned to the user. From a user point of view, the response time is also decreased thanks to this embodiment.

Although illustrative embodiments of the present invention have been described in detail with reference to the accompanying drawings, it is to be understood that the invention is not limited to those precise embodiments and that changes and modifications may be effected therein by those in the art without departing from the scope and spirit of the invention.

# CLAIMS

1. Method for processing a data search based on a complex query for at least one set of combined elements matching search parameters, at least two of the combined elements being of different element types, said method comprising the query processing steps of:

i) obtaining definition data of elements of at least two different element types, said definition data having heterogeneous data structures, said definition data comprising at least one element fare for each element,

ii) converting said definition data into converted definition data sharing one single generic data structure,

iii) receiving the complex query at a search engine,

iv) at the search engine, finding elements of at least two different element types whose converted definition data matches the query and combining said elements, the step of combining comprising:

- associating element fares of said elements,
- building at least one fared set of combined elements based on association of element fares,

v) returning at least one fared set of combined elements.

2. Method according to the preceding claim, comprising the steps of storing the definition data in a repository and of making the search engine access the repository.

3. Method according to any of the preceding claims, the step of combining comprising checking if the association of element fares complies with combinatory rules.

4. Method according to any of the preceding claims, wherein the complex query includes data defining a plurality of components derived from the search parameters and for which at least one element is to be searched for.

5. Method according to the two preceding claims in combination, the step of combining comprising:

- for each component, retrieving a set of matching elements,
- associating matching element fares,
- checking if the association of the matching element fares complies with the combinatory rules,
- if the association complies with the combinatory rules, validating the association of the matching element fares,

- if the association fails to comply with the combinatory rules, invalidating the association of the matching element fares.

6. Method according to the preceding claim comprising repeating the steps of claim 5 with at least a further association of matching element fares.

7. Method according to any of the preceding claims comprising defining a plurality of components derived from the search parameters and for which at least one element is to be searched for, and defining, for each component, at least one element type for which at least one element is to be searched for.

8. Method according to claim 3, comprising:

- executing the query processing steps for a plurality of queries to build associations of matching element fares;,

- building associations of matching elements based on the associations of matching element fares;

- storing the associations of matching elements,

- upon receipt of a request of an end user, retrieving at least one of said associations which matches the request,

- checking if the at least one of said associations complies with the combinatory rules,

- if the at least one of said associations complies with the combinatory rules, validating the at least one of said associations,

- if the at least one of said associations fails to comply with the combinatory rules, invalidating the at least one of said associations,

- returning to the end user the at least one of said associations if it is a validated one.

9. Method according to claim 3, comprising:

- executing the query processing steps for a plurality of queries to build associations of matching element fares,

building associations of matching elements based on the associations of matching element fares,

- checking if the associations of matching elements comply with the combinatory rules,

- for each of the associations of matching elements, if it complies with the combinatory rules, validating said association, and it fails to comply with the combinatory rules, invalidating said association,

- storing the validated associations,

- upon receipt of a request of an end user, retrieving at least one of said validated associations which matches the request,
- returning to the end user the at least one of said validated associations.

10. Method according to any of the preceding claims wherein the definition data comprises fare data, availability data and application rule data.

11. Method according to any of the preceding claims wherein the element types include at least two of:

- air travel element type,
- railway travel element type,
- accommodation element type,
- car rental element type.

12. Method according to any of the preceding claims performed as a result of execution of computer software by at least one data processor, the computer software being stored in a non-transitory computer-readable memory medium.

13. System for processing a data search based on a complex query comprising means for executing the method of any of the preceding claims.

14. System according to the preceding claim wherein the search engine is a low fare search engine.

15. The system as in any of the two preceding claims, comprising computer software stored in a non-transitory computer-readable memory medium that is executed by at least one data processor that comprises part of the system.

16. Computer program product storing a computer program comprising instructions adapted to perform the method of any of claims 1 to 12.

## Claims

1. A method for processing a search query, the method comprising:

obtaining definition data comprising data elements from first and second data repositories at a processor, wherein the data elements of the definition data from the first data repository have a first data structure that differs from a second data structure of the data elements of the definition data from the second data repository, the first data structure comprises a first travel product type and one or more first element fares, and the second data structure comprises a second travel product type different from the first travel product type and one or more second element fares;

converting the definition data from the first data repository and the definition data from the second data repository with the processor into converted definition data sharing one single generic data structure;

storing the converted definition data in a database;

after the converted definition data is stored in the database, receiving the search query at a search engine;

querying the database with the search engine to find the data elements of the first and second travel product types for which the converted definition data matches the search query;

associating the one or more first element fares and the one or more second element fares of the data elements of the first and second repositories based upon compliance with combinatory rules;

building at least one fared set of data elements of the first travel product and the data elements of the second travel product based on the association of the element fares that comply with the combinatory rules; and

for a first association of first and second element fares:

determining if the first association complies with at least one combinatory rule;

if the first association is determined to comply with the at least one combinatory rule, validating the first association; and

if the first association is determined to not comply with the at least one combinatory rule, invalidating the first association such that the first association is excluded from the at least one fared set.

2. The method of claim 1 wherein the query includes data defining a plurality of components derived from search parameters and for each of which at least one data element is to be searched for.

3. The method of claim 1 comprising:

defining a plurality of components derived from the search parameters and for which at least one element is to be searched for; and

defining, for each component, the first and second travel product types for which at least one data element is to be searched for.

4. The method of claim 1 wherein the definition data comprises fare data, availability data, and application rule data.

5. The method of claim 1 wherein the first and second product types include at least two of: air travel element type, railway travel element type, accommodation element type, and car rental element type.

6. A computer program product comprising:

a non-transitory computer-readable memory medium; and

a computer program stored on the medium, the computer program comprising instructions configured to be executed by at least one data processor to process a search query, the instructions configured to:

obtain definition data comprising data elements from first and second data repositories, wherein the data elements of the definition data from the first data repository have a first data structure that differs from a second data structure of the data elements of the definition data from the second data repository, the first data structure comprises a first travel product type and one or more first element fares, and the second data structure comprises a second travel product type and one or more second element fares;

convert the definition data from the first data repository and the definition data from the second data repository into converted definition data sharing one single generic data structure;

store the converted definition data in a database;

after the converted definition data is stored in the database, receive the data search query at a search engine;

query the database with the search engine to find the data elements of the first and second travel product types for which the converted definition data matches the search query;

associate the element fares for the converted definition data that matches the search query based upon compliance with combinatory rules;

build at least one fared set of data elements of the first travel product and the data elements of the second travel product based on the association of the element fares that comply with the combinatory rules; and

for a first association of first and second element fares:

determine if the first association complies with at least one combinatory rule;

if the first association is determined to comply with the at least one combinatory rule, validate the first association; and

if the first association is determined to not comply with the at least one combinatory rule, invalidate the first association such that the first association is excluded from the at least one fared set.

7. The computer program product of claim 6 wherein the query includes data defining a plurality of components derived from search parameters and for each of which at least one data element is to be searched for.

8. The computer program product of claim 6 wherein the instructions are further configured to:

define a plurality of components derived from the search parameters and for which at least one element is to be searched for; and

define, for each component, the first and second travel product types for which at least one data element is to be searched for.

9. The computer program product of claim 6 wherein the definition data comprises fare data, availability data, and application rule data.

10. The computer program product of claim 6 wherein the first and second product types include at least two of: air travel element type, railway travel element type, accommodation element type, and car rental element type.

11. A system for processing a data search query, the system comprising:

a first data repository including definition data comprising a plurality of data elements with a first data structure, the first data structure including a first travel product type and one or more first element fares;

a second data repository including definition data comprising a plurality of data elements with a second data structure that differs from the data structure of the data

elements of the first definition data, the second data structure including a second travel product type and one or more second element fares;

a database; and

at least one processor coupled with the first and second data repositories, wherein the at least one processor is configured to:

obtain the definition data from the first repository and the definition data from the second data repository, convert the definition data from the first data repository and the definition data from the second data repository into converted definition data sharing one single generic data structure, and store the converted definition data in the database;

receive the data search query, query the database to find the data elements of the first and second travel product types for which the converted definition data matches the search query, associate the element fares for the converted definition data that matches the search query based upon compliance with combinatory rules, and build at least one fared set of data elements of the first travel product and the data elements of the second travel product based on the association of the element fares that comply with the combinatory rules; and

for a first association of first and second element fares:

*determine if the first association complies with at least one* combinatory rule;

if the first association is determined to comply with the at least one combinatory rule, validate the first association; and

if the first association is determined to not comply with the at least one combinatory rule, *invalidate the first association such that the first association* is excluded from the at least one fared set.

12. The system of claim 11 wherein the query includes data defining a plurality of components derived from search parameters and for each of which at least one data element is to be searched for.

13. The system of claim 11 wherein the at least one processor is further configured to:

define a plurality of components derived from the search parameters and for which at least one element is to be searched for; and

define, for each component, the first and second travel product types for which at least one data element is to be searched for.

14. The system of claim 11 wherein the definition data comprises fare data, availability data, and application rule data.

15. The system of claim 11 wherein the first and second product types include at least two of: air travel element type, railway travel element type, accommodation element type, and car rental element type.