

IMPROVEMENTS IN OR RELATING TO A SEARCH ENGINE AND ASSOCIATED METHOD

Description

5 Field of the invention

The present invention relates to a method and system for providing an improved search engine, particularly but not exclusively for enabling the processing of travel fares in a generic manner from a variety of sources on a Global Distribution System (GDS).

10 Background of the invention

A global distribution system (GDS) is the de facto standard in the airline industry and publishes fares through a variety of different bodies, such as an airline tariff publishing company (ATPCo), SITA, official airline guides (OAG) or appropriate lookalikes. The "standard carrier" is a carrier that
15 publishes its fares to any of the above-mentioned bodies and is linked into the GDS. Amongst these carriers there are full service carriers (FSC) and also airlines categorized as low-cost carriers (LCC).

Ticketless access (TLA) carriers work in a different way, as they do not publish fares but instead price journeys themselves and manage the
20 whole chain from fare quotation to passenger name record (PNR) management. Typically these types of carriers do not produce a paper ticket.

The travel and leisure market has changed significantly in recent times as new providers of content have emerged, mostly through the Internet. This
25 has resulted in a large diversity of methods and tools. In addition, the new content is often not available through any medium other than that

designed by the provider of that content, particularly where that provider is relatively new in the marketplace. As a result, end users wishing to find the cheapest air fares must visit a rapidly increasing number of different websites. There is still no guarantee however that this will result in
5 actually finding the most cost-effective solutions for a number of different reasons. Firstly, there is an irreconcilable difference between inputs expected by each provider which makes it difficult for the user to select the best search criteria. In addition, the high variability of business models limits the capability to accurately compare equivalent offers. Finally, meta-
10 searches have significant limitations and often do not provide the full chain of travel for a particular journey or product and do not have access to all the content providers.

Figure 1 is a diagram of the current state of the art and demonstrates a typical case when an end-user is searching for a travel or leisure solution
15 from several providers. Here the end-user is searching websites for TLA carriers 103, meta-search websites 102, on-line travel agency (OLTA) websites 100, and airline websites 101 via a FSC website. The OLTA and standard carriers are based on a GDS product (104 and 105). The meta-search websites are based on parsing OLTA 106 and airline 107 websites
20 as well as on GDS products 108. In addition the search websites or the OLTA may have some links to certain TLA websites (109, 110 and 111). This may be achieved by parsing or using a direct link. The GDS is typically connected to a centralized provider (112) such as an ATPCo or an OAG. The TLA websites access content provider from respective TLA
25 providers (113).

The figure 2 shows a state-of-the-art dataflow demonstrating the different data routes depending on the type of carrier. For a TLA carrier 20 the data passes from a TLA user interface to a TLA data manager 200, TLA

solutions 202 and a solutions aggregator 204 in a Low Fare Search (LFS) engine 206. The data from the standard carriers, for example via the FSC or a pioneer carrier (standard carrier 22), passes to a fare and rules databases 208, 210 on the one hand. Whilst on the other hand the flight and availability data passes to flight and availability databases 212, 214. Subsequently, data from the rules and fare databases and the flights and availability databases pass to the universal faring engine 216 and then onto a subsequent FSC and a pioneer solutions database 218 before arriving at a solutions aggregator 204. The format of each different dataflow leads to the creation of a separate recommendation for the TLA carriers and the standard carriers. This is not ideal.

Indeed the prior art leads to certain drawbacks for both the travel vendor and the end-user. In particular the travel vendor has to maintain a number of parallel methods to obtain or book recommendations from different providers. Similarly since there is no vendor capable of offering information relating to all providers, the end-user has to conduct a number of different searches to attempt to compare and contrast differently presented solutions.

Objects of the invention

It is an object of the present invention to overcome at least some of the problems associated with the prior art.

It is a further object of the present invention to provide method and system capable of transforming and formatting data from multiple sources and of multiple natures into one comprehensive data format within, for example, the airline industry.

Summary of the invention

The present invention provides a method and system as set out in the accompanying claims.

- 5 According to one aspect of the present invention there is provided a method of searching for a entity amongst a plurality of entities, wherein all the entities are linked by at least one criterion in common and have a plurality of data formats, wherein each entity comprises a set of data which can be searched by means of criteria entered by a user; and wherein each
- 10 entity has a data format; and further wherein a specific one of the data formats defines a standard record format; the method comprising the steps of: determining the data format of an entity; if the data format of the entity does not match the standard record format, passing the entity to a pre-processing engine; if the data format of the entity does match the
- 15 standard record format, passing standard record format data of the entity to a main processing engine; in the pre-processing engine comparing the entity data format to the standard record format and applying a conversion to the entity set of data; converting the entity set of data to the standard record format based on the conversion to form a converted set of data;
- 20 passing the converted set of data to the main processing engine; in the main processing engine receiving the standard record format entity set of data and the converted set of data; and producing a set of results combining the converted set of data and the standard record format data for selection by a user of a preferred entity.
- 25 The present invention provides a generic fare search engine for the airline industry; for example, thereby reducing costs and infrastructure. The quality of searches for TLA carriers is greatly enhanced since the search engine benefits from fare search options and from improvements on a single generic engine. The present invention allows recommendations

that include a mix of standard carriers and TLA carriers being booked and paid for at the same time. By virtue of the improved engine full integration of TLA and standard carriers is thus achieved.

5 The present invention provides an end-user with travel fares from any travel provider irrespective of the data format and searching method used. This will increase the quality of offers provided to the user. The offers provided will be based on the end-user requests, up-to-date, accurate and readily bookable. In addition, the user interface is comprehensive and user-friendly.

10 The present invention allows transparent and complete integration of existing standard and nonstandard market sectors in the airline industry. This invention leads to the creation of a mixed engine able to treat simultaneously TLA carriers and standard carriers' requests and output a mixed solution within the same timescale and quality as existing search
15 engines. In addition, the business application processing of standard carriers will be able to process TLA carriers with minimal change.

In summary, the present invention solves at least two major problems. The first is to reduce data format multiplicity by enabling integration of standard carrier processes and TLA carrier processes to thereby generate
20 a market open standard that can interact with proprietary standards. A further problem solved is that of the difference in pricing logic can be accommodated and be presented to the user in a manner that allows comparison.

Brief description of the drawings

25 Reference will now be made, by way of example, to the accompanying drawings, in which:

Figure 1 is a diagram of the state of the art configuration for searching for an airline tickets, in accordance with the prior art,

Figure 2 is a diagram of the state of the art dataflow, in accordance with the prior art,

Figure 3 is a diagram of a dataflow, in accordance with an embodiment of the invention,

5 Figure 4 is a diagram of a reverse pricing engine, in accordance with an embodiment of the invention,

Figure 5 is a diagram of the end-to-end system, in accordance with an embodiment of the invention,

10 Figure 6 is a block diagram showing the construction components of an algorithm, in accordance with an embodiment of the invention,

Figure 7 is an example of the process of a reverse pricing engine, in accordance with an embodiment of the invention,

Figure 8 is a block diagram of an enhanced faring engine, in accordance with an embodiment of the invention,

15 Figure 9 is a possible screenshot of the results of a particular business application, in accordance with an embodiment of the invention.

Detailed description of the preferred embodiments

20 Referring to figure 3, a modified dataflow is presented in accordance with the invention. A component or module called the reverse pricing engine 300 is introduced into the flow of TLA data. The reverse pricing engine 300 receives an input of TLA data from the TLA carrier 301 and eventually converts the data to standard data (i.e. data formatted using the de facto standard for travel information which will be described in greater detail

25 below). The TLA data has a specific data format that may change from time to time. The input from the TLA is dynamically updated and is not

stored as such in the reverse price engine, although data is always available. In addition, the universal faring engine 302 is enhanced to be able to take into account specific constraints and rules from the TLA carrier. The universal faring engine is thus capable of creating both TLA
5 and standard carrier recommendations and combining them so that a mixed recommendation 304 can be produced. The reverse pricing engine 300, the enhanced universal faring engine 302, and a mixed recommendation 304 are located in an LFS engine 306.

The data from the standard carriers, for example via the FSC or a pioneer
10 carrier (standard carrier 308), passes to a fare and rules databases 310, 312 on the one hand. Whilst on the other hand the flight and availability data flows to flight and availability databases 314, 316. The data passing to the fare and rules data are typically updated asynchronously five to ten times a day. Subsequently, data from the rules and fare databases and
15 the flights and availability databases pass to the enhanced universal faring engine 302 and then into a mixed recommendation 304.

The reverse pricing engine will now be described in great detail with reference to figure 4. The reverse pricing engine 400 includes
20 standardisation components or modules 402 and fare and rules construction components or modules 404. TLA data 406 is received at the standardisation components and then passed through to the fare and rules construction components. The TLA data includes at least prices, taxes and flight information. The standardisation component analyses the data received from the TLA carrier. This received data can then be formatted
25 using proprietary values (e.g. unnamed passenger type) or may not fit the expected format (e.g. no fare basis, currency different from requested currency etc.). The standardisation component may include a catalogue of values including at least a standardisation passenger (PAX) type 408, a standardisation currency block 410, a standardisation fare basis block

412, and a standardised flight block 414. Further blocks or modules may be included as required in order to enable TLA data to be standardised to that dictated by the de facto standard.

The fare and rules construction component includes a records
5 construction component or module 416 which converts or maps the
standardised elements into predetermined records shown generally at
418. These records 418 are a de facto standard of fare categories in the
airline industry. The input data will at anytime map onto the records in a
predetermined way. This mapping or conversion is stored in the fare and
10 rules construction component. If the data format of the TLA data changes,
the mapping or conversion rules will also change and be updated. The
records 418 produced will depend upon the input data and required
standardisation. Based on the de facto standard the present example
shows the output from the records construction component to be fares
15 420, records 0, 1, 2 and 3 (422, 424, 426, 428 respectively); records S1
and S2 (430, 432 respectively); and UIX files 434. In addition, flights 436
are output based on the standardisation of flights in the standardised flight
block 414. Each of the outputs is associated with the predetermined
parameter. The predetermined parameters include the following: fares;
20 fare properties; carriers; bookings; fare names; date; flight application;
passengers; surcharges; discounts, comment; combinations; rules; taxes;
fees; flight; and any other appropriate parameter. The predetermined
parameters are also defined as part of the de facto standard and are each
linked with one of the records.

25 It should be noted that the records exhibit a property that makes them
versatile, this property is known as "stringing". A string is made of several
records which may be of different natures. Each string points to another
such that the validation of the first is made through the validation of the
other. Alternatively, the strings may qualify one another such that the

validation of the first record is a pre-requisite of the validation of the next. In a preferred embodiment stringing will be used to build records that reconcile the large amount of TLA data received as the input with the de facto standard parameters.

5 Figure 5 is an equivalent diagram to figure 1 in which the invention as implemented is shown. It can clearly be seen that the amount of traffic is significantly reduced by implementing the invention. In addition, the GDS becomes a central component in each and every communication ensuring the data relating to any flights can be accessed thereby. Details of the
10 various communications are not described in detail although the reference numbers are equivalent to those in figure 1 where appropriate. It can be seen that communication from the OLTA and meta search to the TLA website and TLA provider in figure 1 are avoided. Instead communications 500 and 502 from the GDS to the TLA provider replace
15 these.

Figure 6 shows a flow diagram of the algorithm developed to build records from the standardised "plain" data. There are three main layers required in order to build the appropriate records. These three layers included the general algorithm 600, specific records processes 602 and a rules
20 manipulation library 604. The general algorithm layer contains the logic of the algorithm itself. The record specific layer contains logic specific to particular records. The record manipulation library contains all routines useful in manipulating the records.

Each of the layers can further be split into four main stages some of which
25 occur in all layers and some of which occur in only certain layers. The four main stages include generate basic record 606; guaranteed pricing 608; exceptions treatment 610; and generation and storage of the final records 612.

The generation of the basic record is a fundamental stage and the form the record takes impacts the convergence of the algorithm and the quality of the final resultant record. The basic record is an initial simple record which can then be the basis of the final record. The formation of the basic record is generally carried out by extracting the most representative data from the input standardised TLA data. This is accomplished by means of knowing the format of the TLA data at any given time. With this knowledge, it is possible to convert fields such as price, cost, taxes etc into equivalent records. Once the basic record is determined the next stages are carried out. All data entries are considered in sequence to ensure that the TLA data is converted into appropriate records. The stage of generating the basic record includes a generate general record step 614 and an initialised record tree step 616 in terms of the general algorithm. In terms of the specific records processes layer this includes a generate general record step 618 and a record creator step 620. The rules manipulation library layer can then interact with the initialised record tree step 616 and the record creator step 620 to provide a record creation service 622.

Table 1

#	Cities	Flights	Passengers	Surcharge
1	NCE – LON	0098	ADT	120
			CHD	120
			INF	120
2	NCE – LON	0100	ADT	100
			CHD	70
			INF	100
3	NCE – PAR PAR – LON	0055	ADT	150
		0075	CHD	75
			INF	50

Table 1 shows three record entries relating to specific flights from the same departure point and the same arrival point. A flight number is then included along with an identification of surcharge for different types of passengers. The records include other data, such as the fare, booking
5 codes, carriers, etc, but for clarity purposes these are not displayed. Nice to London are the two common criteria for all data and are generally entered by the user at the start of the search. Accordingly, each set of data will have the criteria in common.

An example of the result of this generation of the basic record stage can
10 be done with table 1. The simplest case is record #1, in which all surcharges are €120, regardless of passenger. In other words a surcharge for a flight from Nice to London will be nominally priced at €120.

The next stage of the process is a guaranteed pricing stage. This stage produces a complete and standard pricing of the currently selected TLA
15 data entry. This is accomplished by using a guaranteed pricing engine which analyses the records but then ignores the known components. In the pricing engine there are fields which contain parameters or results. The pricing engine will ignore its own results and will calculate results from other parameters. The pricing engine will then check the results that are
20 calculated and stored. Each record item in the record tree is priced in step 624 by accessing pricing services 626 in the rules manipulation library.

It can clearly be seen that records #2 and #3 are not the same as record #1 and constitute so-called exceptions. The next phase of the algorithm then goes on to generate exceptions in case of difference as will be
25 described below. As previously identified €120 is selected as the base result or basic record for surcharges between Nice and London, since this is the simplest case (surcharges for all passenger types are the same). Other records show different surcharges overall and also vary for different passengers and are then treated as exceptions. With reference to figure 6

the exception is identified in step 628 and then analysed as follows before being added to the exception to records tree 630.

The analysis for calculating the exceptions within the reverse pricing engine is described with reference to figure 7. Figure 7 relates to record 3 (428) in figure 4 which deals with surcharges. Initially a basic record 700 of €120 is selected for comparison with a pricing itinerary. Comparing pricing itinerary #1 identifies a surcharge of €120 for all passengers. Thus pricing itinerary #1 matches the basic record. Looking next at pricing itinerary #2 this relates to flight number 0100 702. Based on this flight number a comparison is made with the basic record 700 and no match is made. As a result, an exception is created : this exception 706 in figure 7 shows that for flight 0100 child surcharges are €70 706 and other surcharges are €100 708. This exception line (706, 708) is stored in the records based on the flight number. Referring to figure 6, this takes place by means of record creator 632 and record creation service 634. For pricing itinerary #2 in respect of surcharges the unique identifier is selected as the flight number 0100. This identifier linked with surcharges cannot be used for any other pricing itinerary in order to form an exception, as there would be no way of differentiating exceptions if there were not a unique identifier. The unique identifier is selected in each case based on the situation and the circumstances of the records.

With reference to pricing itinerary #3 a different unique identifier must be identified. Pricing itinerary #3 relates to a trip which makes a stop. Pricing itinerary #1 and pricing itinerary #2 both have no stops. Accordingly, the process of comparing pricing itineraries with a basic record can be carried out as shown generally at 710 in figure 7. The first question 712 is to determine the number of stops for a flight. If number of stops equals zero 714 step 702 is repeated to determine whether the pricing itinerary matches the base record 700 or the exception line 706, 708. If the

number of stops is greater than zero a new exception 716 is created. The new exception represents pricing itinerary #3 and relates to a greater number of stops than zero with an infant surcharge of €50, a child surcharge of €75 and an adult surcharge of €150. This new exception 716
5 can also be stored in the exceptions part of the database.

Once all the pricing itineraries in a given set have been completed the final record is generated as shown in figure 6, 636. In this case the given set is surcharges between Nice and London. It will be appreciated that producing exceptions will be carried out in respect of all sets of data
10 relating to the de facto standard for categories 418 in figure 4. Surcharges is part of record 3, along with discounts, combination rules etc. Record 2 relates to date, flight application, passengers, etc. The final result of the processing will be a base record for each de facto standard fare category along with a set of exceptions based on the TLA data that does not match
15 the base record for that particular de facto standard fare category. This is dynamically updated as new TLA data is received and stored as standard records for each element of data in the TLA data. A set of results is built for a particular trip or flight such that the results include all the different price options available for the particular trip or flight. The standard record
20 contains all TLA data for a particular enquiry.

The standard records representing TLA data are still not suitable for injection into existing faring engines, as they would not give correct results. This is due to the fact that traditional standard carrier engines produce one ticket per recommendation even if that recommendation
25 relates to more than one airline. As there is no agreement between the TLA carriers and the standard carriers for interlining, a recommendation mixing TLA carriers and standard carriers on a single ticket would not be feasible. Multi ticket standard carrier engines produce several tickets for one recommendation looking for a best combination in terms of pricing.

Mixing TLA carriers and the standard carriers on such an engine would not produce the required partitioning of tickets due to the specific logic required for each carrier. This again would not be feasible.

The present invention addresses this problem by use of an enhanced
5 faring engine 800 which will now be described in greater detail with
reference to figure 8. Enhanced faring engine 800 makes changes to the
traditional standard carrier faring engine to produce a faring engine able to
mix the TLA carriers and standard carriers within one single
recommendation. The enhanced faring engine receives information
10 relating to TLA flights 802 and standard carrier flights 804. A module
referred to as an itinerary partitioning module 806 combines the standard
carriers and TLA carrier flights propositions that cover a requested
itinerary. The itinerary is split into components, each being specific to a
type of carrier. The pricing of each component is then determined so that
15 the split and the pricing of each part can be seen separately. A mixed
pricing logic module 808 create price core units (PU) are both standard
carriers and TLA carriers by applying construction rules that are specific to
each. A mixed combination logic module 810 organises the price of the
units into a complete unit, which covers the complete itinerary for both the
20 TLA carriers and standard carriers. This is accomplished by a
combination of specific rules. A complete coverage for a particular
itinerary is referred to as a "Meta CU". An algorithm referred to as the
Meta CU Front 812 analyses all the created Meta CU data and keeps the
best ones to create a mixed recommendation. Recommendations that are
25 unreasonable (for example, too many legs, too expensive, too slow etc)
are discarded first. Then when a predetermined number of
recommendations is reached these are displayed. As the process
progresses the next result is selected based on the previous result to

determine the best price available for each part or component of the particular trip.

The Meta CU algorithm is critical to reduce the amount of processing that would normally be required in order to create a mixed recommendation.

5 For example taking a journey made of three segments and considering the pricing options of three TLA carriers and all the standard carriers (all the standard carriers account as one set of data due to the fact that the standard carrier information is all in the appropriate format) and there are essentially four carriers each with three different segments.

10 Mathematically this equates to $4^3 = 64$. This means there are 64 possible combinations from which the best recommendations need to be identified. The Meta CU algorithm reduces the number of recommendations until they reach a certain or predetermined value. Once the best recommendations have been identified and combined they are output 814.

15 The output 814 from the enhanced their engine for a journey from Nice to London and then London to Nice could be as shown in figure 9.

Figure 9 shows a selection of different recommendations of flights from Nice to London and return flights from London to Nice. The table includes departure and arrival location along with departure and arrival times. In
20 addition the airline has identified along with the flight number. Finally the ticket price is identified along with a total price for the return journey. Entry 900 proposes an outward flights using TLA1 and an inbound flight using TLA2. The total cost is determined to be €160. Entry 902 proposes an outward flight of two legs using an FSC and return flights also the two legs
25 using the same FSC. In this case the total price is the same as the ticket price.

A person skilled in the art will understand that some or all of the functional entities as well as the processes themselves may be embodied in

software, or one or more software-enabled modules and/or devices or in any combination thereof.

This invention has been described with reference to the purchase of tickets in the travel environment. However, it will be appreciated that the invention may apply to other environments, for example comparing prices for other goods and services. The various goods, services, products, etc. may be referred to as an entity. A user may wish to search for an entity matching certain criteria (e.g. data stored in the records). In the example described above, the entity is a flight ticket; the criteria are the start and finish locations and the results of the search are possible flights available for the given criteria including details thereof such as price, numbers of stops, and any other appropriate information.

It will be appreciated that this invention may be varied in many different ways and still remain within the intended scope of the invention.

Claims

1. A method of searching for an entity amongst a plurality of entities, wherein all the entities are linked by at least one criterion in common and
5 have a plurality of data formats, wherein each entity comprises a set of data which can be searched by means of criteria entered by a user; and wherein each entity has a data format; and further wherein a specific one of the data formats defines a standard record format; the method comprising the steps of:
- 10 - determining the data format of an entity;
- if the data format of the entity does not match the standard record format, passing the entity to a pre-processing engine;
- if the data format of the entity does match the standard record format, passing standard record format data of the entity to a main processing
15 engine;
- in the pre-processing engine comparing the entity data format to the standard record format and applying a conversion to the entity set of data;
- converting the entity set of data to the standard record format based on the conversion to form a converted set of data;
- 20 - passing the converted set of data to the main processing engine;
- in the main processing engine receiving the standard record format entity set of data and the converted set of data;
- producing a set of results combining the converted set of data and the standard record format data for selection by a user of a preferred entity.
- 25
2. The method of claim 1, further comprising carrying out the steps for multiple entities from multiple sources.

3. The method of claim 1 or claim 2, wherein the method of converting entity data to a standard record format comprises the steps of:
- identifying a first basic record value corresponding to a field of data in the standard record data format from one of the sets of data for a particular entity;
 - comparing the first basis record value with the record value for an equivalent field of data of a subsequent entity;
 - if the first basic record value matches the record value for the equivalent field of data of the subsequent entity using the first basic record value in the field for the subsequent entity;
 - if the first basic record value does not match the record value for the equivalent field of data of the subsequent entity creating an exception record for the field for the subsequent entity.
4. The method of claim 3, further comprising repeating the steps for all subsequent entities to there by generate a first basic record and a list of exceptions for each record field.
5. The method of claim 3 or claim 4, further comprising repeating the steps for each record files for the or each entity.
6. The method of any preceding claim, further comprising presenting a set of results to the use of selection and purchase of a preferred entity.
7. A system for searching for a entity amongst a plurality of entities, wherein all the entities are linked by at least one criterion in common and have a plurality of data formats, wherein each entity comprises a set of data which can be searched by means of criteria entered by a user; and wherein each entity has a data format; and further wherein a specific one

of the data formats defines a standard record format; the system comprising:

- a module for determining the data format of an entity;
- if the data format of the entity does not match the standard record format,
5 passing the entity to a pre-processing engine;
- if the data format of the entity does match the standard record format,
passing standard record format data of the entity to a main processing
engine;
- wherein the pre-processing engine compares the entity data format with
10 the standard record format and converts the entity set of data to the
standard record format thereby forming a converted set of data;
- wherein the main processing engine receives the standard record format
entity set of data and the converted set of data to produce a combined set
of results including the converted set of data and the standard record
15 format data to enable a user to select a preferred entity.

8. The system of claim 7, wherein multiple entities are searched for from multiple sources.

- 20 9. The system of claim 7 or claim 8, wherein the pre-processing
engine converts entity data to a standard record format by identifying a
first basic record value corresponding to a field of data in the standard
record data format from one of the sets of data for a particular entity to
thereby compare the first basis record value with the record value for an
25 equivalent field of data of a subsequent entity; and
- if the first basic record value matches the record value for the equivalent
field of data of the subsequent entity using the first basic record value in
the field for the subsequent entity;

- if the first basic record value does not match the record value for the equivalent field of data of the subsequent entity creating an exception record for the field for the subsequent entity.

5 10. The system of claim 9, wherein all subsequent entities are compared to thereby generate a first basic record and a list of exceptions for each record field.

10 11. The system of claim 9 or claim 10, wherein each record file are processed for the or each entity.

15 12. A pre-processing engine for use in the system of any of claims 7 to 11, and which converts entity data to a standard record format by identifying a first basic record value corresponding to a field of data in a standard record data format from one of a set of data for a particular entity to thereby compares the first basis record value with a record value for an equivalent field of data of a subsequent entity; and

20 - if the first basic record value matches the record value for the equivalent field of data of the subsequent entity using the first basic record value in the field for the subsequent entity;

- if the first basic record value does not match the record value for the equivalent field of data of the subsequent entity creating an exception record for the field for the subsequent entity.

25 13. A computer program comprising instructions for carrying out the steps of the method according to any one of claims 1 to 6, when said computer program is executed on a computer system.