DESCRIPTION OF THE ALGORITHMS

1. Algorithm I

The algorithm I deals with queries that are equivalents in terms of *abstract services*. This means that the set of *abstract services* of an incoming query is equal to the set of *abstract services* of a query in the query history. The steps below describes the algorithm I:

Step 1**:** Search in the history for previous queries that have exactly the same abstract services included in the incoming query.

Step 2**:** For the set of queries retrieved in the *step 1*, select the queries that have requirements equivalent or more restrict than the incoming query. If there is result for this step, go to *step 4*. Otherwise, continue in the *step 3*.

Step 3**:** For the set of queries retrieved in the *step 1*, which consequently have requirements less restrict than the incoming query, order them according to their timestamp, type (Processed or Reused) and score giving priority to the most recent, processed and higher scored queries. Thus, pick the first one and use it on the *step 7b*.

Step 4**:** For the set of queries retrieved in the *step 2*, select the queries that have requirements equivalent. If there is result for this step, go to *step 5*. Otherwise, go to *step 6*.

Step 5**:** For the set of queries retrieved in the *step 4*, order them according to their timestamp and type (Processed or Reused) giving priority to the most recent and processed queries. Thus, pick the first one and use it on the *step 7*.

Step 6**:** For the set of queries retrieved in the *step 2*, order them according to their timestamp, type (Processed or Reused) and score giving priority to the most recent, processed and higher scored queries. Thus, pick the first one and use it on the *step 7*.

Step 7: Given a query to be reused, identify the compositions produced to this query that have online services and order them according to their score. Pick the first composition, return it to be executed, and go to *step 8*.

Step 7b**:** Given a query to be reused, identify the compositions produced to this query that have online services, which satisfy the requirements, and order them according to their score. Pick the first composition, return it to be executed, and go to *step 8*. [Otherwise, pick the higher scored query, even without covering completely the requirement, return it to be executed if the user agrees according the percentage of requirements that this query covers. I think here could appear the degree of satisfaction that we talked in the last meeting.]

Step 8**:** Save the new query in the history as "reused" query, copying the compositions selected in the *step 7* and the services used by these compositions.

2. Algorithm II

The algorithm II deals with queries that are superset in terms of *abstract services*. This means that the set of *abstract services* of an incoming query is contained in the set of *abstract services* of a query in the query history. The steps below describes the algorithm II:

Step 1**:** Search in the history for previous queries that have a set of abstract services bigger than the incoming query, but containing the abstract services included in the incoming query.

Step 2**:** For the set of queries retrieved in the *step 1*, select the queries that have requirements equivalent or more restrict than the incoming query. If there is result for this step, go to *step 4*. Otherwise, continue in the *step 3*.

Step 3**:** For the set of queries retrieved in the *step 1*, which consequently have requirements less restrict than the incoming query, order them according to their timestamp, type (Processed or Reused) and score giving priority to the most recent, processed and higher scored queries. Thus, pick the first one and use it on the *step 7b*.

Step 4**:** For the set of queries retrieved in the *step 2*, select the queries that have requirements equivalent. If there is result for this step, go to *step 5*. Otherwise, go to *step 6*.

Step 5**:** For the set of queries retrieved in the *step 4*, order them according to their timestamp and type (Processed or Reused) giving priority to the most recent and processed queries. Thus, pick the first one and use it on the *step 7*.

Step 6**:** For the set of queries retrieved in the *step 2*, order them according to their timestamp, type (Processed or Reused) and score giving priority to the most recent, processed and higher scored queries. Thus, pick the first one and use it on the *step 7*.

Step 7**:** Given a query to be reused, identify the compositions produced to this query that have as online services the ones that covers that abstract services of the incoming query, and order them according to their score. Pick the first composition and go to *step 8*.

Step 7b**:** Given a query to be reused, identify the compositions produced to this query that have as online services the ones that covers that abstract services of the incoming query, which satisfy the requirements, and order them according to their score. Pick the first composition and go to *step 8*. Otherwise, pick the higher scored query, even without covering completely the requirement and go to *step 8b*.

***Step 8*:** Given a composition, (i) make its projection including only the services that covers the abstract services in the incoming query; (ii) validate the rewriting; (iii) if the rewriting is valid, return it to be executed and go to *step 9*. Otherwise, pick another composition in the *step 7* (or *step 7b* depending on which *step* called *step 8*) and perform again the *step 8*.

***Step 8b*:** Given a composition, (i) make its projection including only the services that covers the abstract services in the incoming query; (ii) validate the rewriting; (iii) if the rewriting is valid, return it to be executed if the user agrees according to the percentage of requirements that this query covers and go to *step 9*.

***Step 9*:** Save the new query in the history as "reused" query, copying the projection of the compositions selected in the *step 7* that are a valid rewrite of the query and the services used by these compositions.

3. Algorithm III

The algorithm III deals with queries that are subset in terms of *abstract services*. This means that the set of *abstract services* of an incoming query contains the set of *abstract services* of a query in the query history. The steps below describes the algorithm III:

***Step 1*:** Search in the history for previous queries that have a set of abstract services included in the set of abstract services of the incoming query.

***Step 2*:** For the set of queries retrieved in the *step 1*, select the queries that have requirements equivalent or more restrict than the incoming query. If there is result for this step, go to *step 4*. Otherwise, continue in the *step 3*.

***Step 3*:** For the set of queries retrieved in the *step 1,* which consequently have requirements less restrict than the incoming query, order them according to their timestamp, type (Processed or Reused), score and size (number of abstract services) giving priority to the most recent, processed, higher scored, and bigger queries. Thus, pick the first one and use it on the *step 7b*.

***Step 4*:** For the set of queries retrieved in the *step 2,* select the queries that have requirements equivalent. If there is result for this step, go to *step 5*. Otherwise, go to *step 6*.

Step 5**:** For the set of queries retrieved in the *step 4*, order them according to their timestamp, type (Processed or Reused), and size (number of abstract services) giving priority to the most recent, processed, and bigger queries. Thus, pick the first one and use it on the *step 7*.

Step 6**:** For the set of queries retrieved in the *step 2*, order them according to their timestamp, type (Processed or Reused) and score giving priority to the most recent, processed and higher scored queries. Thus, pick the first one and use it on the *step 7*.

Step 7**:** Given a query Q_1 selected in the *step 5*, search for a query Q_2 that complements Q_1 using the query selection process in the algorithm I. Select the compositions of Q_1 and Q_2 that have online services, which satisfy the requirements (depending on the type of the requirements Q_2), and order them according to their scores. Pick the first composition of each query and go to *step 8*.

Step 7b**:** Given a query Q_1 selected in the *step 5*, search for a query Q_2 that complements Q_1 using the query selection process in the algorithm I. Select the compositions of Q_1 and Q_2 that have online services, which satisfy the requirements and order them according to their scores. Pick the first composition of each query and go to *step 8*. Otherwise, pick the higher scored query, even without covering completely the requirement and go to *step 8b*.

Step 8**:** Given two compositions, (i) combine them; (ii) validate the rewriting; (iii) if the rewriting is valid, return it to be executed and go to *step 9*. Otherwise, pick another composition in the *step 7* (or *step 7b* depending on which *step* called *step 8*) and perform again the *step 8*.

Step 8b**:** Given two compositions, (i) combine them; (ii) validate the rewriting; (iii) if the rewriting is valid, return it to be executed if the user agrees according to the percentage of requirements that this query covers and go to *step 9*.

Step 9**:** Save (i) the new query in the history as “reused” query, (ii) the valid rewritings produced by combing the queries selected in the *step 7* and (iii) the services used by these compositions.