5 Web Service Compositions

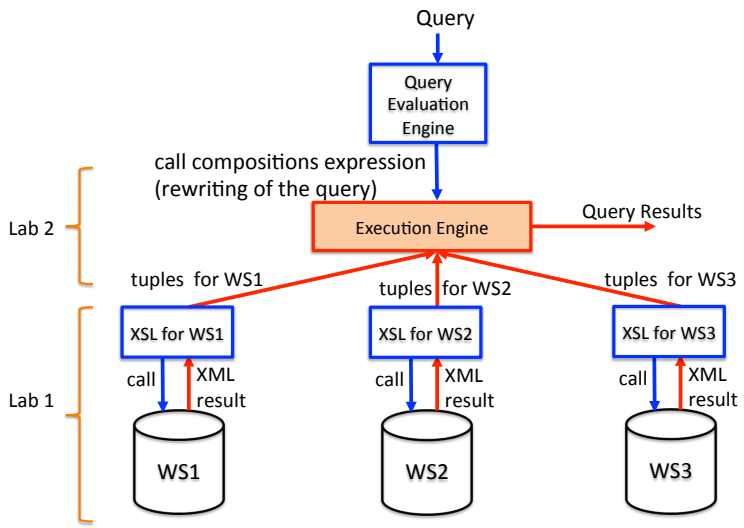

Figure 1: Overview of the Query Evaluation Engine

The goal is to develop an execution engine that is able to execute call composition plans. The algorithm will take as input an expression representing a call composition. The execution engine could be used for instance for executing the plans produced for some given used quert by some query reviewing algorithm. Figure 1 shows overall architecture of the query evaluation engine.

Let's consider again the WS given as example in the last class:

$$\text{getArtistInfoByName}^{iooo}\ (?\text{artistName}, ?\text{artistId}, ?\text{beginDate}, ?\text{endDate})$$

We know that we can count on the transformation function to give transform the results of call results into tuples of the relation getArtistInfoByName. Assume now that, two other WS are defined, namely:

$$\text{getAlbumByArtistId}^{ioo}(?\text{artistId}, ?\text{beginDate}, ?\text{albumId}, ?\text{albumName}) \text{ and}$$
$$\text{getSongByAlbumId}^{iooo}(?\text{albumId}, ?\text{singerName}, ?\text{songTitle}, ?\text{year})$$

Intuitively, a call composition is a conjunctive query consisting of an ordered sequence of atoms. As example, consider the following expression:

$$\text{getArtistInfoByName}^{iooo}(\text{"Frank Sinatra"}, ?\text{id}, ?\text{b}, ?\text{e})$$
$$\#\ \text{getAlbumByArtistId}^{ioo}(?\text{id}, ?\text{b}, ?\text{aid}, ?\text{albumName})$$
$$\#\ \text{getSongByAlbumId}^{iooo}(?\text{aid}, \text{"Frank Sinatra"}, ?\text{title}, ?\text{year})$$

The first expression ($\text{getArtistInfoByName}^{iooo}(\text{"Frank Sinatra"}, ?\text{id}, ?\text{b}, ?\text{e})$) denotes a single function call. The second expression ($\text{getAlbumByArtistId}^{ioo}(?\text{id}, ?\text{b}, ?\text{aid}, ?\text{albumName})$)

denotes a set of function calls, one for each input ?*id* obtained as answer of the first expression. The third expression (getSongByAlbumId^{iooo}(?aid, "Frank Sinatra", ?title, ?year)) denotes a set of function calls, one for each input ?aid.

Definition 1 (Workflow (Plan)) *A workflow is a sequence of function calls. Each argument in the workflow has to be either (1) a constant symbol that appears in Q or (2) a variable.*

The workflow is *admissible* if, for every variable, the first occurrence of the variable in the call sequence is in an *o*-position of a function call. Let us look again at our sample plan above. The plan is admissible because every *i*-argument of a function call is either a constant or has been bound (was given a value) by a previous function call.

Execution. The execution of the expression is left to right, step by step. At each step we join the results of two expressions function calls and obtain a new table. For our example, we have 3 function call expressions, hence we will have two joins. We first compute the results for

getArtistInfoByName^{iooo}("Frank Sinatra", ?id, ?b, ?e)
getAlbumByArtistId^{ioo}(?id, ?aid, ?albumName)

This will lead to a table partialResults1^{oooo}("Frank Sinatra", ?id, ?b, ?e, ?aid, ?albumName). Note that every variable in occurring in the two expressions appears once in the head of the new table. Then, we have to join:

partialResults1^{oooo}("Frank Sinatra", ?id, ?b, ?e, ?aid, ?albumName)
getSongByAlbumId^{iooo}(?aid, "Frank Sinatra", ?title, ?year)

The result is a table: partialResults2^{oooo}("Frank Sinatra", ?id, ?b, ?e, ?aid, ?albumName, "Frank Sinatra", ?title, ?year).

Joins. The first call will result in a (small) table of tuples. This table will provide the inputs for the function calls of the second expression. Hence we have a join between the results of the first calls and the results provided by the second call. A join should also be performed when the same variable is marked as output variable for two different Web calls of a composition. For instance, assume that the WS getAlbumByArtistId returns also the birthday of the artist. Then the call composition

getArtistInfoByName^{iooo}("Frank Sinatra", ?id, ?b, ?e)
getAlbumByArtistId^{ioo}(?id, ?b,?aid, ?albumName)

is equivalent to

getArtistInfoByName^{iooo}("Frank Sinatra", ?id, ?b, ?e)
getAlbumByArtistId^{ioo}(?id, ?b1,?aid, ?albumName)
?b=?b1

Selections. Note that in the third first and in the third expression, a constant ("Frank Sinatra") occurs on the place of an input-output and on the place of an output variable. In both cases, a selection should be performed. The algorithm should select only the tuples where the variable on that position are equal to ("Frank Sinatra").

6 Assignment

Implement an execution engine for call compositions. The execution engine receives as input an expression such as

$$\text{getArtistInfoByName}^{iooo}(\text{"Frank Sinatra"}, ?id, ?b, ?e)$$
$$\# \ \text{getAlbumByArtistId}^{ioo}(?id, ?aid, ?albumName)$$

The engine should be able to

1. parse the expression → get the list of functions to be called (for simplicity, you can use # to separate calls. It's will be easier to parse the expression)
2. execute the calls in the order described by the composition
3. output the results