В этом алгоритме мы имеем древовидную структуру опроса.

Каждый опрос имеет корень - Root.

Узлы строго типизированы. То есть мы имеем 3DGrid, Grid, Single, OpenText, Answer, PredefinedList ...

Для каждого узла устанавливается логический урвень. Например у корня - 0. Длаее Grid3D - 1 и т.д.

У каждого узла, кроме корня есть родитель и лог уровень этого родителя меньше чем чем у этого узла. Кроме predefined list - у них одинаковый.

Например:

1) 3DGrid может содержать Grid, Single

2) Grid может содержать Single, OpenText.

3) Single может содержать PredefinedList, Answer.

4) PredefinedList может содержать Answer.

5) Но Answer уже ничего может в себе содержать.

Есть 3 вида связи в каждом соединении между двумя узлами

а) Строгая - когда два узла связаны

б) Относительная - когда связан хотя бы один из детей этих узлов

в) Отсутствие, в этом случае узел заменяется на пустой

При добавлении связи происходит проверка : если связанны родители, то связи потомков внутри могут быть только между потомками этих родителей.

Связать два узла возможно только если их логические уровни совпадают.

Работа алгоритма :

1) Копирование структуры левого дерева, тем самым на выходе будет два дерева с одинаковой структурой - но правое дерево будет состоять из пустых узлов.

2) Для каждого узла из левого дерева, с которым имеется связь в правом(пустом) дереве вместо пустого вставляем узел из первоначального правого дерева,

с которым связан узел из левого дерева

3) В правом полупустом дереве пытаемся для каждого пустого узла восстановить его, основываясь на первоначальном правом дереве

4) Для каждого узла пытаемся вставить ребёнка между ним и его детьми, если в правом дереве имеется промежуточный узел

При соединении окончательня структура дерева формируется из главного опроса.

При соединении 2 и более деревьев получается один и тот же тип. При соединении получается также древовидная структура.

То есть каждый узел также имеет свой тип. Но в каждом узле лежит структура из узлов разных опросов с соответсвующим типом связи.

In this alhoritm we have a tree structure of surveys - this means that target survey and all source surveys have the same structure - the tree.

All surveys have to be converted to the tree structure. Each survey has a Root.

And all nodes are strongly typed. So, we have a grid3d, grid, single, opentext, answer, predefinedList and so on.

For each node we specify a logic level. For example: for Root we should always set a zero level. For grid3d - 1, MultiGrid - 2, Grid - 3 and so on.

All nodes of course have a parent and the logic level of each node is less than logic level of his parent.

This rule is true for all types of node except predefinedList. The value of logic level means that whether node can contain another node or not.

If the value of logic level of first node is less than value of logic level second node - this means this that first node can contain second node

and therefore the second node can not contain the first node.

Let's take some examples

1) grid3d can contain Grid and single,

2) grid can contain Single, OpenText

3) Single can contain PredefinedList and answer

4) PredefinedList can contain answer and PredefinedList

5) But answer can not contain PredefinedList, or something else

There are 3 kinds of connecion between nodes. They are :

1) Strong - this kind of connection occures when 2 nodes are bound

2) Relative - this kind of connection occured when there are at least 2 nodes which are bound and each node is child of these nodes which have relative connection

The first node is out of target tree and the second is out of second tree;

3) Absent - this kind of connection occurs when 2 nodes are not bound with themselves.

When adding a connection, more precisely, when binding two nodes the validation happens:

If parents of these nodes are bound then the childs of these nodes can be bound only with themselves.

The descendants of node out of target tree can be bound only with descendants of node out of source tree.

And of course we can bind two nodes only if their values of logic levels are the same.

The point of alhoritm:

1) The first moment of this alhorimt is - coping of structure of target tree. So - in the end we will have the two equal tree in their structure

2) In this step we have 2 trees with the same structure and they look like one tree. Let's name this complex tree as DoubleTree.

Each node of this DoubleTree is 2 nodes, the first one is out of target tree and the second node is empty node. let's consider that the empty node is future place for node, which will be bound with target node. And for all nodes out of DoubleTree we do the next:

If first node which is out of the target tree, is bound with some node which is out of the source tree, we have to assign the source node to second node which is in the DoubleNode. And in the end of this step we should have the structure of DoubleTree, which are so:

a) Each node of DoubleTree have 2 nodes with strong connection, because these two nodes were bound. Some of nodes can have empty second nodes, because these two nodes were not bound.

3) In this step we have to repair second nodes of this DoubleTree. We still have the original source tree with which we bind the target tree. So for each empty second node from DoubleTree we can get the parent from original source tree. and in this way we can assign the parent which is out of source tree to second empty node from DoubleTree. Of course we can do so, only if the logic level of first node from DoubleTree is equal to logic level of parent, which we are trying to assign to the second empty node from DoubleTree.

4) For each node from DoubleTree we are trying to insert another node, which structure is: The first node is empty and the second node is out of the source tree. And the second node contains the second nodes of children, which are children of this DoubleNode. And the first or second nodes of parent of this DoubleNode can contain the second node of inserted DoubleNode.

After connecting this trees, the end structure of tree will be formed, basing on the structure of target tree. When binding two or more trees between themselves in the end the combined surveys are in the tree structure. So each node has his type and logic level. But in each node we have node from target tree and set of nodes from source trees. They all have the same logic level and the same type. But some of them can be empty.