7^{th} international conference for informatics and information technology (CIIT 2010)

KNOWLEDGE DELIVERY ORDER IN AWARENESS

Velimir Graorkoski Marjan Gusev
Institute of Informatics,
Univeristy of "Sts. Kiril and Metodij"
Skopje, Macedonia

Abstract

A specific problematic task in Awareness [1] model of adaptive learning is the order of which the learning aggregates are delivered to the learner. In this paper, several strategies for ordering of the learning concepts, realized in Awareness, are explained, in which partially are used some of the algorithms for graph traversal (mainly Breadth First Search and Depth First Search), as well as some other techniques.

Introduction

Awareness model of adaptive learning is a realization of the key terms from the theory of the adaptive learning, like concept [2], adaptive learning environment (ALE) [3], adapted learning environment (IALE) [3] etc. In Awareness, the concepts organized in ALE refer to a wider mass of learners, but for a concrete learner an IALE is formed consisting only of those concepts from ALE for which it is assumed that it is not familiar with. The problem of the learning order of the concepts starts in the moment when IALE is fully formed, i.e. it is determined which concepts from ALE will be included in it, as well as the connections between them. The representation with oriented graph of a single IALE, according to the connections between the concepts without learning priority defined (Def. 1) is shown on Fig. 1.

Def 1: learning priority of concepts

> The value which is assigned to every concept learning candidate, which is used to determine the order of the way the concepts are delivered to the learner, will be called a learning priority.

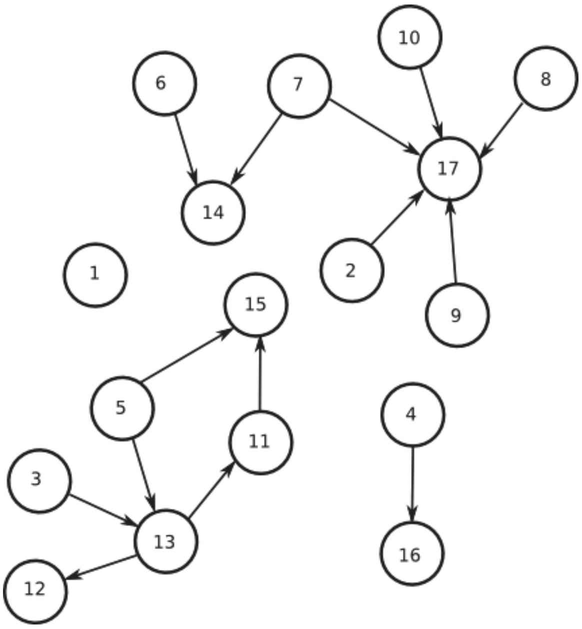


Figure 1. Example of IALE with 17 concepts, represented by an oriented graph

No matter if the concepts in IALE have a priority or not, always the first concept to be offered to the learner is one of those which does not need pre knowledge to be learned, i.e. for which there isn't another concept from IALE which brings knowledge to it. The problem of finding a way to determine the learning order of the concepts begins when there is more than one such candidate in IALE without defined priorities, or expressed in graph theory terminology, vertices which are roots for an oriented non weighted graph [4].

Specifically, the whole problem of finding a learning order will be shown on an IALE instantiated with Awareness, which is implemented as a relational database, with an E-R diagram on Fig. 2. According to it, the relational tables Functioni consist of the concepts with i number of other concepts as pre knowledge. Therefore, the concepts from 1 to 10 from Fig. 1. belong to Function0 table, the concepts 11, 12 and 16 in Function1 table etc.

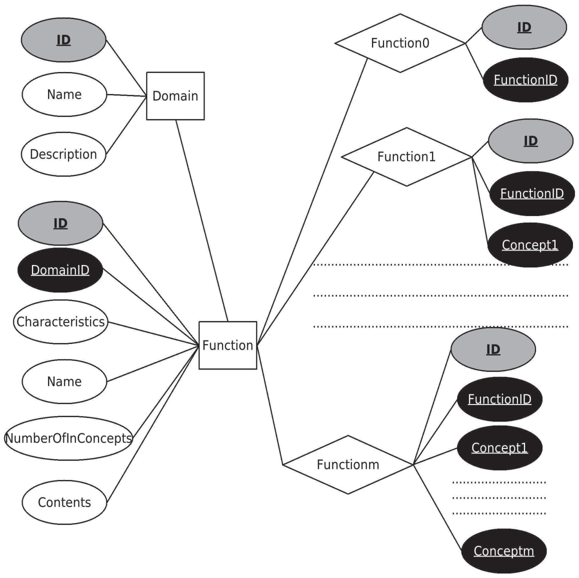


Figure 2. E-R diagram of IALE, instantiated with Awareness

An example of IALE consisting of 17 concepts, realized in Awareness is shown on Fig. 3. A check if a single IALE is correctly instantiated can be executed with the formula (1), where the number of rows in the central table Function, equals to the sum of the rows of all n tables relations Functioni. For the example on Fig. 3. using (**1**) we can confirm IALE's correctness, because it is clear that 17 = 10 + 3 + 3 + 0 + 0 + 1.

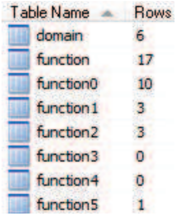


Table Name	Rows
domain	6
function	17
function0	10
function1	3
function2	3
function3	0
function4	0
function5	1

Figure 3. IALE realized as a relational database[1]

SELECT COUNT(*) FROM Function = (SELECT COUNT(*) FROM Function0) + (SELECT COUNT(*) FROM Function1) + ... + (SELECT COUNT(*) FROM Functionn). (1)

In addition, based on a specific example of IALE in Awareness, we will explain several methods like, Breadth First Learn – BFL and Depth First Learn - DFL, of finding a way to sort the concepts into an array which will mark the order of learning. The columns Characteristics and Contents from the central table does not have to be treated at all, as well as the column ID from the tables relations. Also the table Domain will not influence the introduced methods.

I Preparation for learning from IALE

Before we refer to the methods for determining the learning order of the concepts from IALE, we will give an example of IALE's database, according to the diagram on Fig. 4, so that we can give a better explanation of them.

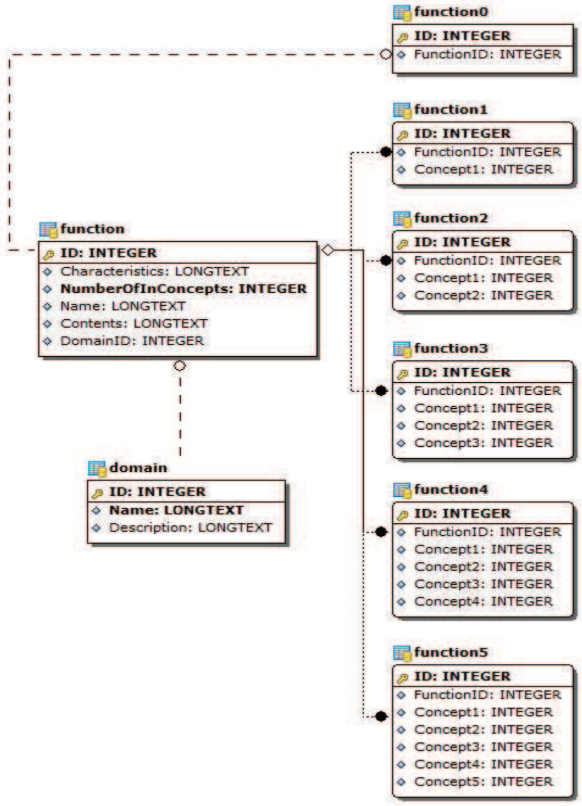
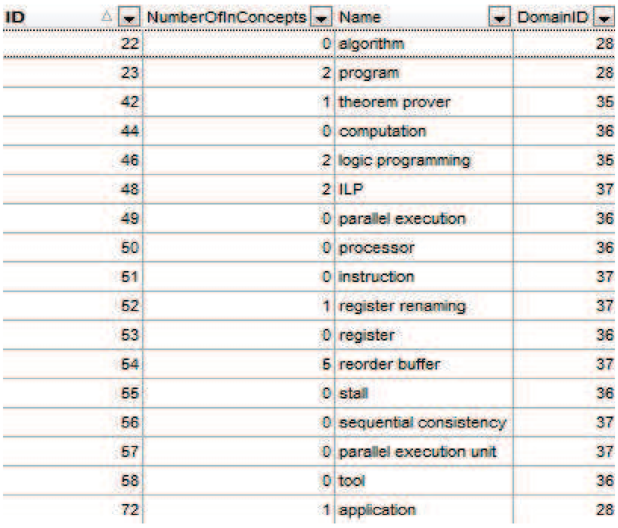


Figure 4. Diagram of IALE's database

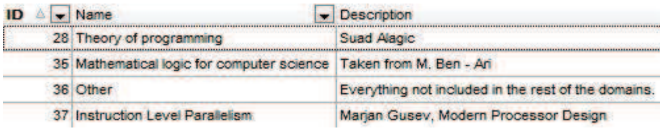
The example of IALE on Fig. 4. contains 5 tables relations, which means that there are concepts gaining pre knowledge from maximum 5 other concepts. On Fig. 5. contents of the table Function is shown, together with the values of the important columns, where in the column DomainID using the identification numbers, the background of every concept to a certain domain is marked, in this case 4 of them, shown on Fig. 6.

[1] Although tables Function3 and Function4 are empty, they exist in the database because in a certain phase of the instantiation of IALE they happen to contain at least one concept (concretely the one which belongs to the table Function5)



ID	NumberOfInConcepts	Name	DomainID
22	0	algorithm	28
23	2	program	28
42	1	theorem prover	35
44	0	computation	36
46	2	logic programming	35
48	2	ILP	37
49	0	parallel execution	36
50	0	processor	36
51	0	instruction	37
52	1	register renaming	37
53	0	register	36
54	5	reorder buffer	37
55	0	stall	36
56	0	sequential consistency	37
57	0	parallel execution unit	37
58	0	tool	36
72	1	application	28

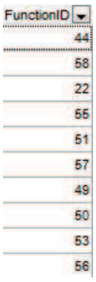
Figure 5. Example of the contents of the table Function



ID	Name	Description
28	Theory of programming	Suad Alagic
35	Mathematical logic for computer science	Taken from M. Ben - Ari
36	Other	Everything not included in the rest of the domains.
37	Instruction Level Parallelism	Marjan Gusev, Modern Processor Design

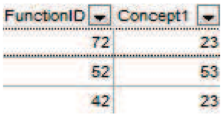
Figure 6. Example of the contents of the table Domain

The contents of the tables relations is shown on Fig. 7., Fig. 8., Fig. 9. and Fig. 10. There are the connections between the concepts in IALE, according to the identification number.



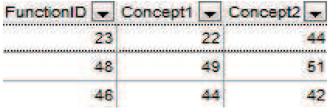
FunctionID
44
58
22
55
51
57
49
50
53
56

Figure 7. Example of the contents of the table Function0



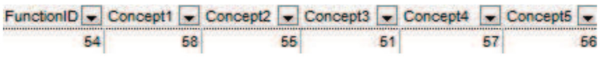
FunctionID	Concept1
72	23
52	53
42	23

Figure 8. Example of the contents of the table Function1



FunctionID	Concept1	Concept2
23	22	44
48	49	51
46	44	42

Figure 9. Example of the contents of the table Function2



FunctionID	Concept1	Concept2	Concept3	Concept4	Concept5
54	58	55	51	57	56

Figure 10. Example of the contents of the table Function5

There is no need for examples of the contents of the tables Function3 and Function4, because according to the table Function, there aren't any fields from the column NumberOfInConcepts with value 3 or 4.

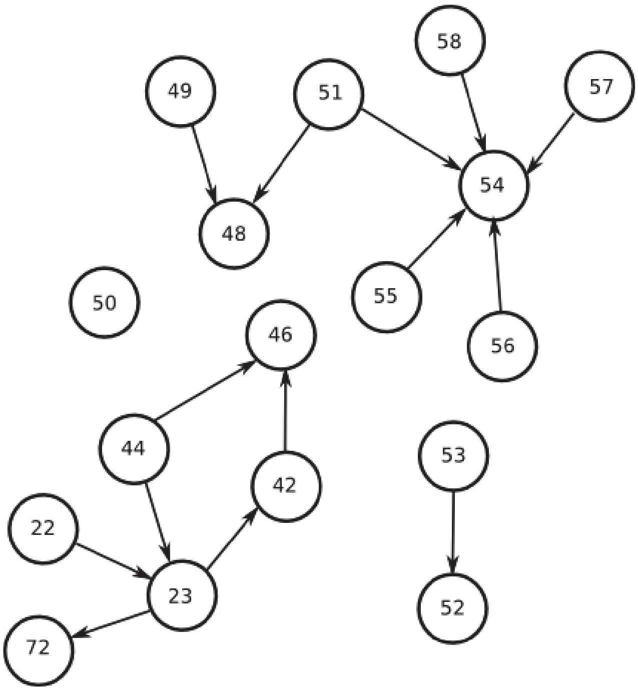


Figure 11. Representation of IALE by oriented graph based on its db instance from the previous examples

In order to explain the methods of learning order easily, it is good to show the following example of IALE's realization, with oriented graph on Fig. 11, where the vertices appropriate to the concepts are enumerated with the identification number from the Function table.

II BFL

The breadth first learn strategy - BFL is based on the algorithm for breadth traversal (BFS) [4] in graphs. First we will examine a case without learning priority. The decision which concept will be the first to be delivered for learning is made according to the defined level inside the graph in which it is situated. The level in which the concept is situated is determined by the length of the longest path starting from one of the root vertices i.e. the concepts which do not need a pre knowledge to be learned. According to that, the roots themselves will belong to the first level – level 0, the concepts with the length of their longest path equal to 1 will belong to the level 1 etc., appropriately for every other concept. The division in levels for our example of IALE will be shown on Fig. 12., where it is clear that the concept “logic programming“ with path length 1 from the concept “computation“, can not belong to level 1 because there is a path with length 3 which leads through the concepts “program“ and “theorem prover“.

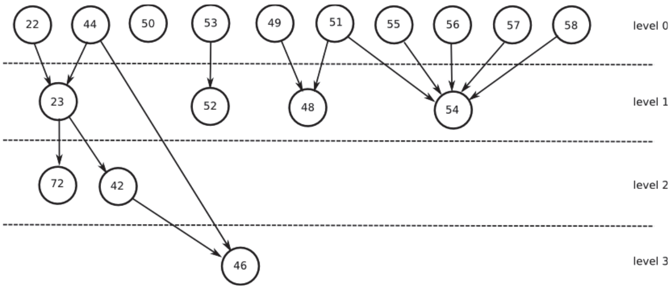


Figure 12. Division by levels of the IALE's graph

With the help of BFL the learning order of the concepts is formed – first, the concepts from level 0 are delivered to the learner, after that from level 1, level 2, and in the end from level 3. Here are three possible arrays:

- 58 57 56 55 53 51 50 49 44 22 54 52 48 23 72 42 46;
- 22 44 49 50 51 53 55 56 57 58 23 48 52 54 42 72 46;
- 50 53 49 55 56 57 58 51 22 44 52 48 54 23 42 72 46;

When learning priorities are assigned to the concepts, then BFL will form the array in the same way going from the first to the last level, except in the same level the concepts with higher priority values will be delivered before the concepts with lower priority values. Fig. 13. shows our example of IALE with assigned priority values.

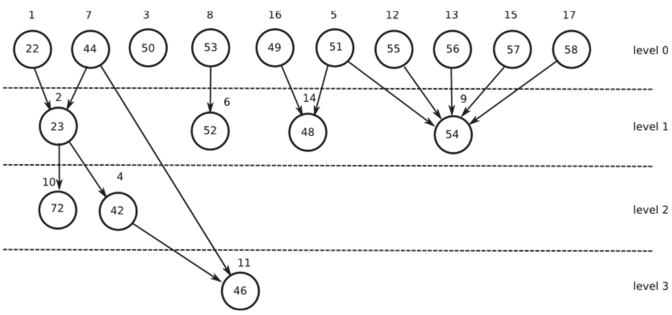


Figure 13. Division by levels of the IALE's graph with assigned priorities

According to the example given on Fig. 13., the learning order of the concepts using BFL, will be the following:

- 58 49 57 56 55 53 44 51 50 22 48 54 52 23 72 42 46;

If one concept which is situated in a higher level has higher priority than another concept situated in a lower level, then the concept from the lower level is delivered before the concept from the higher level, no matter the priority value. This can be easily seen from the previous order where the concepts “ILP“ and “application“ with priority values 10 and 14 respectively, although have a higher priority than the concepts “register renaming“ and “processor“ with priority values 6 and 3 respectively, because of the levels to which belong, they are delivered to the learner later.

BFL does not have a significant difference compared to the standard BFS algorithm, if the learning dependencies expressed by the introduction of the priorities are not calculated, which is proved by the resulted arrays – they are always equal to one another.

III DFL

Unlike BFL, the method based on the algorithm for depth traversal in graphs (DFS) [4] which will be explained in addition, does not use layer division according to the longest path, but it forms the learning order of the concepts according to a different strategy. It starts by finding a path from a certain concept with maximum length so that all concepts found on the path do not have other concepts that bring knowledge to them and also not delivered for learning yet, i.e. they are not previously traversed by the method. According to this traversal by depth method or depth first learn (DFL), possible arrays which can be formed from our example on Fig. 11. are the following:

- 22 44 23 72 42 46 50 53 52 49 51 48 55 56 57 58 54;
- 22 44 23 42 46 72 50 49 51 48 55 56 57 58 54 53 52;
- 51 55 56 57 58 54 49 48 22 44 23 72 42 46 53 52 50;

If priorities are assigned to the vertices, then no matter the priority value, the vertices candidates for forming a longest path are chosen and a unique array is formed, of course in case all of the priority values assigned to the IALE's concepts are not equal. If we consider the example on Fig. 13., then the only possible array of concepts to be delivered on a learner, according to DFL is the following one:

- 58 57 56 55 51 54 49 48 53 52 44 22 23 42 46 72 50;

From the previously formed array it can be seen that even though the concept “parallel execution“ with priority 16 has higher priority value than the concept “reorder buffer“ with priority 9, the first one to be delivered will be the second concept because it is a candidate for forming a longest path from the concept which was the first to bring knowledge to it - “tool“.

The priority has influence only when we traverse from a concept which brings knowledge to more than one concept. That would be the case with the concepts “program“ and “logic programming“ towards which a knowledge is brought from the concept “computation“ (only if the connection between the concepts “program“ and “theorem prover“ does not exist, as shown on Fig. 14.) and here the higher priority value which which is assigned to the concept “logic programming“ decides the choosing of the next concept to be the part of the array.

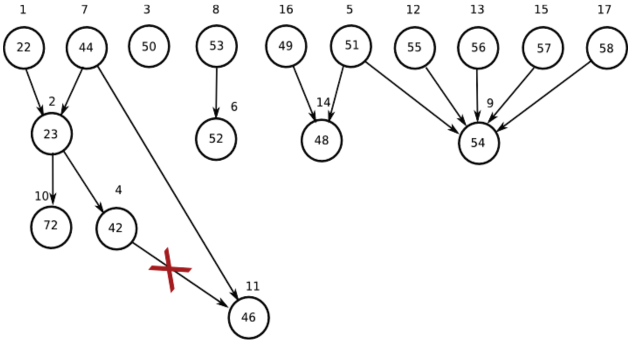


Figure 14. Example of IALE's oriented graph with priority values and without marked levels

In Awareness we use one modification of DFL which functions the same way except when a terminal concept[2] is traversed. If that comes, no matter if there are candidates for forming a longest path, the next concept to be delivered will be one of the concepts which do not need pre knowledge. One such array for our example is the following, formed by this modification:

– 22 44 23 72 50 53 52 49 51 48 55 56 57 58 54 42 46;

DFL differs from DFS according to the way of forming the learning order of the concepts because in DFS the traversal goes always in depth without limitations according to the check whether some of the concepts needs a pre knowledge from other concepts from the so far extracted learning array, till the terminal concept.

IV Comparisons and conclusion

In conditions when in the role of a learner appears an intelligent system capable of long term memorizing of the concepts, it is not a problem which method for learning order of IALE's concepts will be used, because this learner's characteristic guarantees us that it will not “заборави“ any of the delivered concepts. According to that, its learning efficiency will not change.

However in case of human as a learner, the choice of the strategy will influence the whole efficiency of the adaptive learning. The human forgets, so we do not have a guarantee that some of the concepts which are delivered by Awareness will remain memorized during the learning process, especially in case the of a large number of concepts. According to this and also from everyday examples related to the humans' learning where it is easier consequentially or in short intervals to learn knowledge areas which are closely related. In this case it means that it is easier to learn the concepts inside the one of the groups of concepts (coincidentally, here they are grouped by the domain in which they belong) on Fig. 15. than to “jump“ from one concept of one group to another concept from another group.

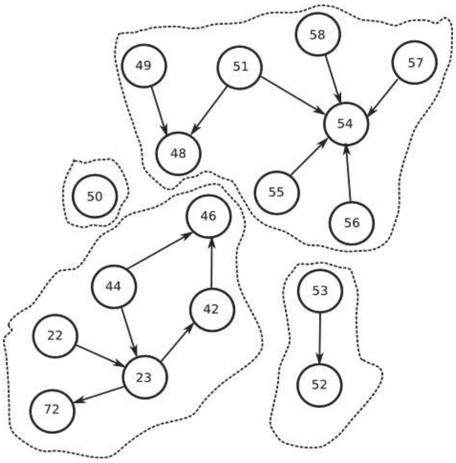


Figure 15. Example of IALE's oriented graph with marked groups of closely related concepts

Considering that reason, we will give an edge to DFL and its modifications over BFL, because in that way, there is a great chance for efficient and full learning of the groups of closely related concepts and until the learning of the whole group is finished, the concepts from the other such group can not be delivered yet.

Based on the cases when priority values are assigned to the concepts, there is no reason for comparison because with the introduction of the learning priority itself, the learning efficiency is altered which is hard for the ordering strategies to change it.

References

[1] V. Graorkoski and M. Gusev, *Model of adaptive e-learning*. Institute of Informatics, Skopje, 2009, pp. 78–87

[2] Concept which can not provide knowledge anymore i.e. It does not bring knowledge to any other concept

[2] A. Paramythis and S. Loidl-Reisinger, *Adaptive learning environments and e-learning standards*. Johannes Kepler University, Linz, Austria, 2004

[3] V. Graorkoski and M. Gusev, *Representation of the Learning Mechanism in Adapive Learning*. CiiT2007, Bitola 21-25.01.2007, (eds. Kusakatov, Zdravkova, Jovanov), Skopje, 2008, pp. 92-112.

[4] D. Jungnickel, *Graphs, networks and algorithms*. University of Augsburg, 1998

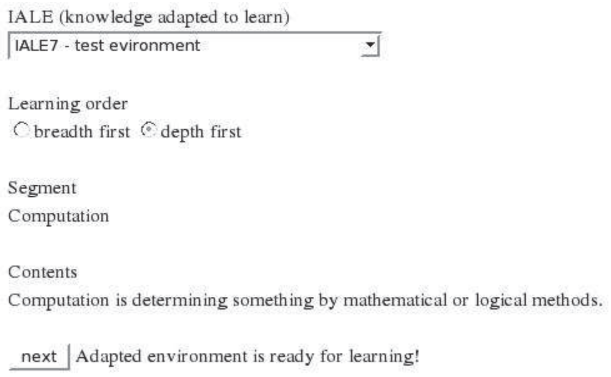
Appendix: Matrix of IALE and form for adaptive learning in Awareness

In this appendix we will present the IALE's matrix which helps us during the realization of BFL and DFL during the adaptive learning in Awareness, as well as the form through which the learning happens.

The matrix of IALE is adequate to the adjacency matrix [4] in oriented graph, with the difference in introduction of 0 row and 0 column in order to store the concepts' identifiers. First, through the class IALE [1] a list of the identifiers of all concepts in IALE is made and it is used for filling the values of the 0 row and 0 column of the matrix, which is previously allocated with 0s. Afterwards for every value of the matrix from the 0 column, again with the help of IALE class, the in-concepts for the concept whose identifier equals to the current value of the column, are chosen. For every identifier of those concepts a value 1 is assigned to that element in the matrix which belongs to the concept's row and the in-concept's column according to the value of the 0 row and 0 column. Following the example of IALE on Fig. 11., with this method the matrix of IALE is formed, shown with the formula (2).

$$\begin{array}{cccccccccccccccccc}
0 & 22 & 23 & 42 & 44 & 46 & 48 & 49 & 50 & 51 & 52 & 53 & 54 & 55 & 56 & 57 & 58 & 72 \\
22 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
23 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
42 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
44 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
46 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
48 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
49 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
50 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
51 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
52 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
53 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
54 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
55 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
56 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
57 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
58 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
72 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array} \tag{2}$$

Why is it necessary to form the matrix of IALE, when the connections between the concepts are already defined in IALE's database itself and the finding of a learning order can be executed directly according to a certain strategy? BFL, DFL and their modifications have NP complexity [4] and for a large number of concepts the result time will be long. If the fact that the calls to the database have a negative effect on the time efficiency is added, then it is normal to try to speed the process. Since we cannot find replacements with less complexity for BFL, DFL and their modifications, we can find a way to encircle the multiple calls to the IALE's database. With the forming of the matrix we can make only one set of calls to the database and the rest of the future operations will happen using the matrix values i.e. in the memory.



IALE (knowledge adapted to learn)
IALE7 - test evironment

Learning order
○ breadth first ● depth first

Segment
Computation

Contents
Computation is determining something by mathematical or logical methods.

Adapted environment is ready for learning!

Figure 16. Form for adaptive learning in Awareness

After all the steps for realization of the model of adaptive learning [1] are executed, it is time to explain the form for the final interaction with the learner. The form for adaptive learning (Fig. 16.) of Awareness provides learning environment as a choice for a learner from a set of instantiated environments, based on a previous testing, and then an overview of the concepts one by one, during which the learning time for a concept is qualified by the learner itself.

What happens at the background of this form? The choosing of the learning environment from the list is in fact making a choice of the modified kernel [1] of IALE, from which the instantiation is being completed till the full IALE. According to the choice of the ordering method (BFL or DFL i.e. modification of DFL), immediately after the complete instantiation of IALE, the first concept of the learning array is displayed. When the learner itself feels ready to learn a new concept, it will activate the next concept in the array and that's how the learning continues till the last concept in the array.