UDC 37.018.43
Original Scientific Paper

Interactions Validation Methods for Training Resources Control Engine Development

Jan Piecha *jan.piecha@polsl.pl*
Informatics Systems Department of Transport
Faculty of Transport, Silesian University of Technology

Marcin Bernaś *marcin.bernas@polsl.pl*
Informatics Systems Department of Transport
Faculty of Transport, Silesian University of Technology

Abstract

The training courseware complexity proper selection is one of the most difficult factors looking from an intelligent application engine development. The application needs individual settings, the most relevant for the application structure matching to the users' individual expectations. What is more, the obtained structure allows controlling dynamically the application within a time it is used. The application units description with their controlling functions allow joining the database components into individual composition of the courseware. The paper introduces several aspects of distance learning resources development, fulfilling the demanding assumptions of the interactive training units.

Keywords: MAMS, LMS, ITS, e-learning, distance learning

1. Introduction

The electronic courseware units, with their organisation composition principles, fulfilling the theoretical relationship and unification factors [1], [4] were defined in many papers, like [6], [7]. They also were implemented in applications controlling engines of Learning Management Systems (LMS) [17], [18], [19].

These training applications (e-courses) are supported by interactive platforms working in accordance with pedagogical training models as the introduced one, discussed in works [3], [8]. They provide the applications with methodological fundamentals, assigned as: linear presentation sequences, alternative trees, and blocks of sequences or complex selected graphs. The majority of training courses manage the smallest fundamental parts (frames, SCO – Sharable Content Object) [5]. Despite the chosen theory or standard, the application current path selection usually proceeds in accordance with the scheme, introduced by block diagram in Figure 1.

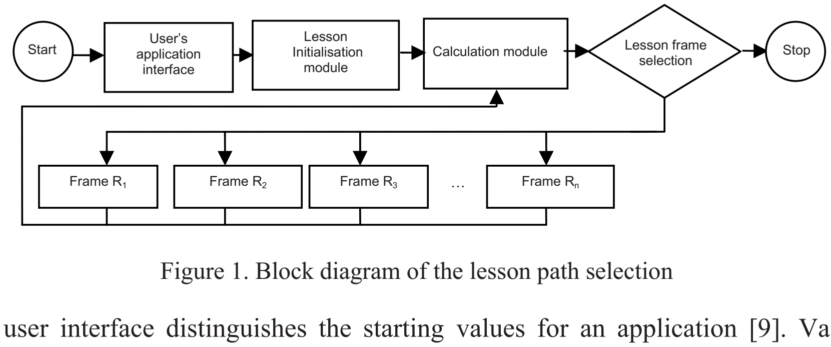


Figure 1. Block diagram of the lesson path selection

The user interface distinguishes the starting values for an application [9]. Values, obtained with interaction facilities, are based on user's preferences (profile). Alternative solution uses several introductory questions (pre-tests) that estimate the data of the courseware starting level; corresponding to the user's knowledge level [9].

Majority of the LMS platforms [10], [11], [12], [13] estimate knowledge increase from single frame by simple measure expressed in percentage form 0% to 100% or by the word set. Simple measures for the single, multiple choice as well as single fill-up format are satisfying for an electronic training system. However one can find many examples where these data-collecting models are not satisfying at all. The presentation frame expresses more sophisticated content, where traditional interactions are not effective enough for defining the user's knowledge.

In Figure 2 simple interactions script based algorithm, with the answers evaluations, was introduced [2].

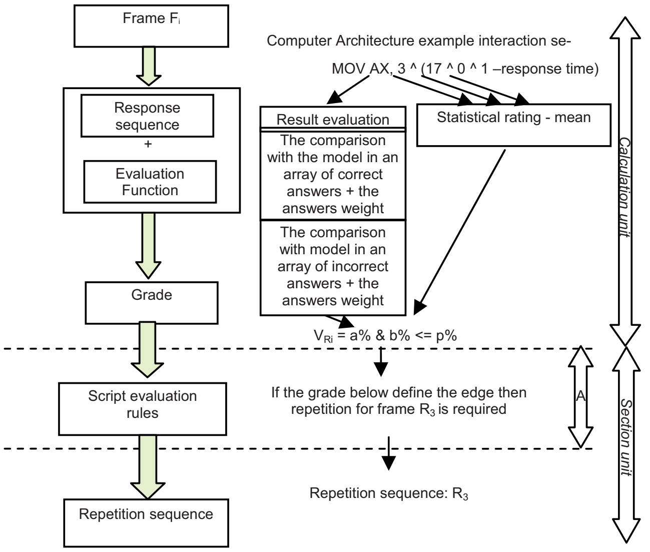

Frame F_i
Computer Architecture example interaction se-
MOV AX, 3 ^ (17 ^ 0 ^ 1 –response time)
Response sequence
+
Evaluation Function
Result evaluation
The comparison with the model in an array of correct answers + the answers weight
The comparison with model in an array of incorrect answers + the answers weight
Statistical rating - mean
Calculation unit
Grade
V_{Ri} = a% & b% <= p%
Script evaluation rules
If the grade below define the edge then repetition for frame R_3 is required
A
Section unit
Repetition sequence
Repetition sequence: R_3

Figure 2. Block diagram of lesson current path selection algorithm

The calculation unit evaluates the user's interaction feedback. The obtained results are compared with the defined requirements for presentation content steering and for the application repetition loops [14].

The application frame mode is set to one – for still active frames and to zero for the frames excluded from the proceeded presentation sequence. The frames mode is set into the selection module in its settings array.

The block diagram presented in Figure 3 introduces the proposal for more complex evaluation procedure of the user's interaction. The extended evaluation algorithm uses the structure of knowledge-content database, for this classification criterions development. The solution was applied in Multimedia Applications Management Shell (MAMS) in its control layer [2], [3].

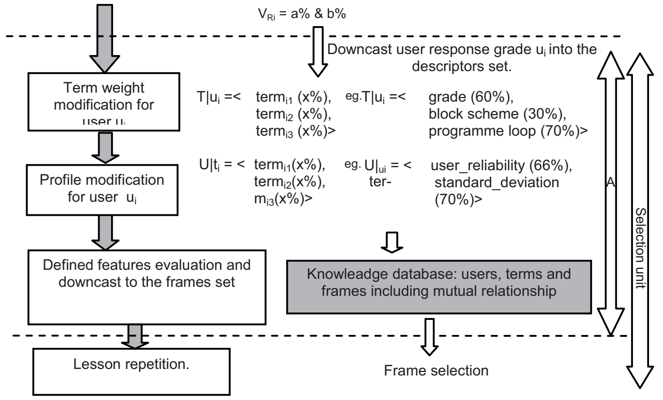

V_{Ri} = a% & b%
Downcast user response grade u_i into the descriptors set.
Term weight modification for user u_i
$T|u_i$ =< $term_{i1}$ (x%), $term_{i2}$ (x%), $term_{i3}$ (x%)>
eg. $T|u_i$ =< grade (60%), block scheme (30%), programme loop (70%)>
Profile modification for user u_i
$U|t_i$ = < $term_{i1}$(x%), $term_{i2}$(x%), m_{i3}(x%)>
eg. $U|_{ui}$ = < ter- user_reliability (66%), standard_deviation (70%)>
A
Selection unit
Defined features evaluation and downcast to the frames set
Knoweadge database: users, terms and frames including mutual relationship
Lesson repetition.
Frame selection

Figure 3. The control flowchart through the knowledge database

The application structure proposal, discussed in this paper, modifies controlling processes, including the application (knowledge) database. For the application functionality improvement the database structure is divided into the three objects' classes:

- Users; with their profiles (abilities and skills level),
- E-content; with the applications frames (single units) identifiers,
- Terms; recording the user's knowledge descriptors.

Two types of the classes are available in majority of a training process control (defined as Learning Management Systems - LMS). The paper discusses the characteristic features of the above classes, with their additional descriptors (as in Figure 3), complementing the structure of an Intelligent Tutoring System (ITS).

The above features are defined as relations of F functions (Figure 4); expressing the function distinct mutual dependency of the database components; within one or two object's belonging to so called layers. The integrity of structure is achieved by relations shown in Figure 5. The above relations require graphs assignments that allow defining links between the data single units (frames) and more complex applications units, as lessons and courses.

The directed multigraph [16], [17], [18] was used as a base for the application structure definition:

$$G = (V, E) \tag{1}$$

where:

$V = \{v_1, v_2, ..., v_n\}$, is a set of vertices representing the system objects (O),

$E = \{e_1, e_2, ..., e_n\}$, is a set of edges describing the relations and their functions (F).

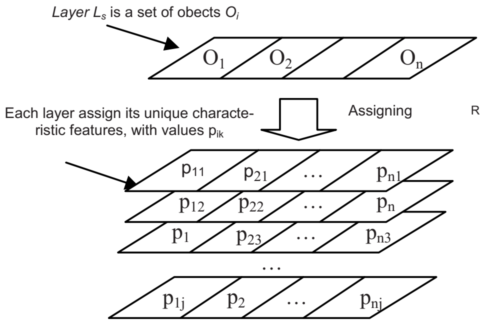


Figure 4.Graphical layers definition

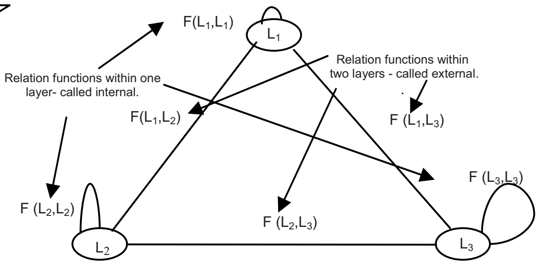

F(L_1,L_1)
L_1
Relation functions within one layer- called internal.
Relation functions within two layers - called external.
F(L_1,L_2)
F (L_1,L_3)
F (L_3,L_3)
F (L_2,L_2)
F (L_2,L_3)
L_2
L_3

Figure 5. References between layers

The vertices of the directed multigraph are divided into three separate sets; according to the defined layers (L):

$$V = R \cup T \cup U \tag{2}$$

$$R \cap T = R \cap U = T \cap U = \phi \tag{3}$$

where:

R - defines the finite set of objects representing single e-content (frame) of the application:

$$R= \{r_1, r_2,…., r_k\}$$

T - defines the finite set of objects; in the frames described by a semi-natural language:

$$T= \{t_1,t_2,….,t_m\}$$

U - is the finite set of objects, representing the users:

$$U = \{u_1,u_2, …,u_n\}$$

The layers R, T, U relationship was defined as:

$$
\begin{aligned}
&V_L = \{v_1, v_2, ..., v_n\}\\
&C_L = \{c_1^L, c_2^L, ..., c_k^L,\}\\
&P_{c_L} = \forall c_j^L \; P_{cl} = \{p_1, p_2, ..., p_m\}\\
&F^V : V_L \times C_L \rightarrow P_{c_L}\\
&\lambda : C_L \rightarrow k : k = [0,1]
\end{aligned}
\tag{4}
$$

where:

v_i - the object (vertices) in the layer L,

L - distinct layer of objects and features,

C_L - the L layer of a finite set of attributes,

P_{c_L} - the set of attributes values,

F^V - the default function, assigning the value p_{cj} of objects v_i and feature $c_{j,}$

λ - the quality function, influence on evaluation and lesson selection process (default = 1) of specified feature.

Each layer is assigned by number of standardised metric features, where its values are normalised by following functions:

$$
p_{C_jVx} = \frac{F^V(v_x, c_j)}{\max\limits_i (F^V(v_i, c_j))}
\tag{5}
$$

or

$$
p_{C_iV_x} = \min(1, \frac{F^V(v_x, c_j)}{\max(P_c)})
\tag{6}
$$

The equation (5) is used when the maximal value of function is defined. Otherwise, based on series of results for given feature, the maximal value is selected (equation 6). Function returns the value within the range [0,1]. The definition is simplified to the following equation (7):

$$
\forall c_{Lj} \in C_L \;, F^V \; : V \times C \rightarrow \{p_C : p_C \in [0,1]\}
\tag{7}
$$

The multi-graph assigns an edge for the ordered pair of vertices:

$$
G : e_i \Rightarrow V \times V
\tag{8}
$$

Relation of Cartesian square product of the layer is called internal relation; otherwise it is an external one. Edges (9) are defined similarly; also the layer:

$$
\begin{aligned}
&E_{L'\times L''} : \{(v_i, v_j) : (v_i, v_j) \in L'\times L''\},\\
&C_{L'\times L''} = \{c_1^{L'\times L''}, c_2^{L'\times L''}, ..., c_k^{L'\times L''},\}\\
&P_{c_{L'\times L''}} = \forall c_j^{L'\times L''} \; P_{cL'\times L''} = \{p_1, p_2, ..., p_m\}\\
&F^E : E_{L'xL''} \times C_{L'xL''} \rightarrow \{p_i : p_i \in P_{C_{L'xL''}}\}\\
&\lambda : C_{L'xL''} \rightarrow k : k = [0,1]
\end{aligned}
\tag{9}
$$

where:

$E_{L'\times L''}$ -edge for an ordered pair of vertices,

$\boldsymbol{F}^E$ -default function, assigning value p_{cj} for edge e_i and the feature $c_{j,}$

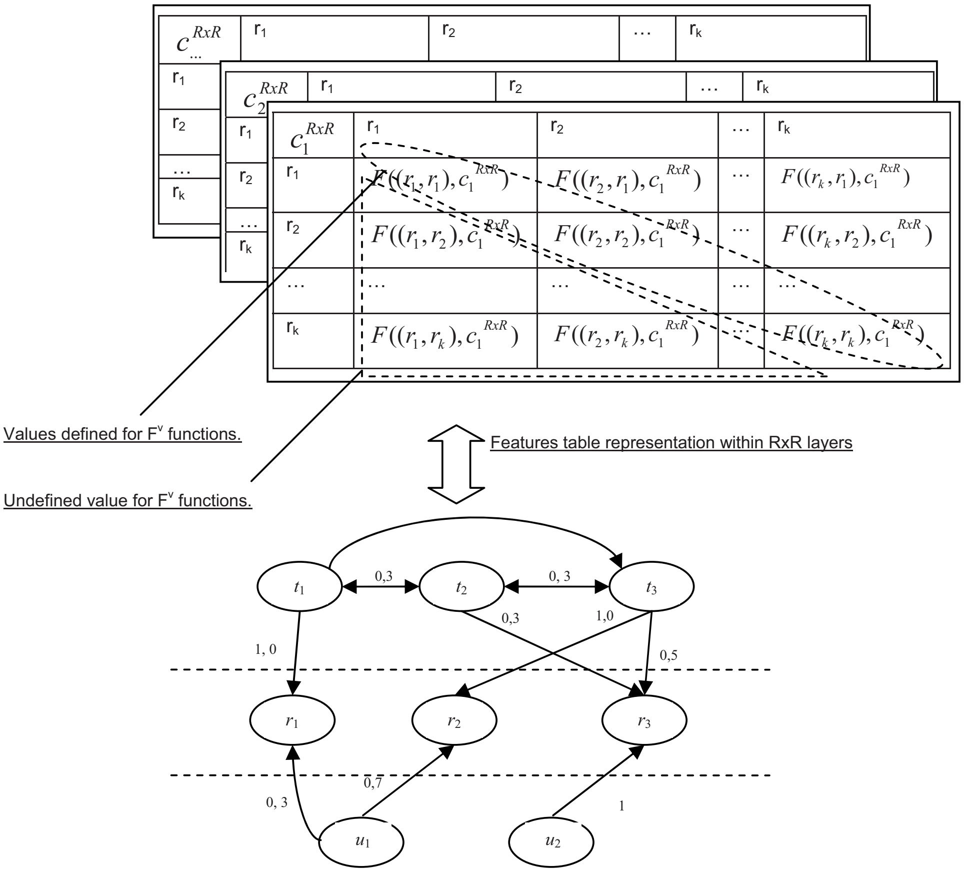
$C_{L'xL''}$ -finite set of features (attributes) for the Cartesian product (layers L'xL''),

$P_{C_{L'xL''}}$ -values set for distinct attributes that belong to the range [0, 1],

λ -quality function, defining evaluation results for the lesson selection (default = 1).

Both functions, F^V and F^E are the Cartesian products. Equation 10 expresses the functions generalisation procedure.

$$\begin{aligned}
&\forall v_i, v_j \in L', E = \{(v_i, v_j) : (v_i, v_j) \in L' \times L'\},\\
&for \quad i = j, F^E : E_{o_i \times o_i} \times C \equiv F^V : V_{o_i} \times C \to \{p_i : p_i \in P_{C_L}\}\\
&for \quad i \neq j, F^E : E_{o_i \times o_j} \times C \to \phi
\end{aligned} \tag{10}$$



c_1^{RxR}	r1	r2	…	rk
r1	$F((r_1,r_1),c_1^{RxR})$	$F((r_2,r_1),c_1^{RxR})$	…	$F((r_k,r_1),c_1^{RxR})$
r2	$F((r_1,r_2),c_1^{RxR})$	$F((r_2,r_2),c_1^{RxR})$	…	$F((r_k,r_2),c_1^{RxR})$
…	…	…	…	…
rk	$F((r_1,r_k),c_1^{RxR})$	$F((r_2,r_k),c_1^{RxR})$	…	$F((r_k,r_k),c_1^{RxR})$

$c_{\dots}^{RxR}$
c_2^{RxR}
r1
r2
…
rk
Values defined for F^v functions.
Undefined value for F^v functions.
Features table representation within RxR layers
t_1
t_2
t_3
0,3
0, 3
0,3
1,0
1, 0
0,5
r_1
r_2
r_3
0,7
0, 3
1
u_1
u_2

Figure 6. The structure representation.

Function F^E (representation of F^V) for two different objects is undefined (empty set) and for the same objects ($v_i = v_j$) is returning the value F^V for v_i vertices. The range ([0,1]) of the function values, simplifies the fuzzy understanding conclusions algorithms. The graphical representation of a definition is presented on Figure 6. The graph features are defined within the layers descriptors.

2. The characteristic features definition and their functions

2.1 The layer T description in semi-natural language

The T layer features were defined by the RDF standard implementation [20], [21]. Terms layer (T) defines the semi-language word syntaxes:

root	is a base of the word without prefix or suffix,
prefix	is predicting the root,
suffix	is the root end.

The function is a pre-defined word type, in the specified language. Abstraction level is based on the author's grade results. Description, contains the information unit, expressing the type, content or additional comments (not evaluated), where $F_{description}$=0.

The T features are words, in the syntactic algorithm, matching the given word with comparison patterns (in Polish and English languages).

Relations within T are extracted from the thesauruses and from the syllabuses specifications [15] with the following relations: Previous, Next, IsPartOf, Has Part, IsBasisFor, Requires, IsRequiredBy, Broader, Narrower, Related, Use; synonym of an object.

Values of the above features are defined by an expert or they are imported into the application database from thesauruses or syllabuses:

$$F : E \times C \rightarrow \left\{ \begin{array}{c} 1 : exist(\ C|E) \\ 0 : don't\ exist(\ C|E) \end{array} \right\} \tag{11}$$

and

$$F : E \times C \rightarrow \left\{ \begin{array}{c} p : exist\ \ (C|E, p) \\ 0 : don't\ exist(C|E, p) \end{array} \right\} \tag{12}$$

where:

$C|E$ – the given feature edge,

p – weight of the relation defined by the thesaurus.

The T layer allows defining a key idea (descriptors) of the lesson. Next step concerns connections of layers (R) with application frames finding. They are related to: the application part, the required unit, broader or narrower content descriptors (features).

The T layer defines the fundamental structure of the application. The example layer relations were introduced in Figure 7.

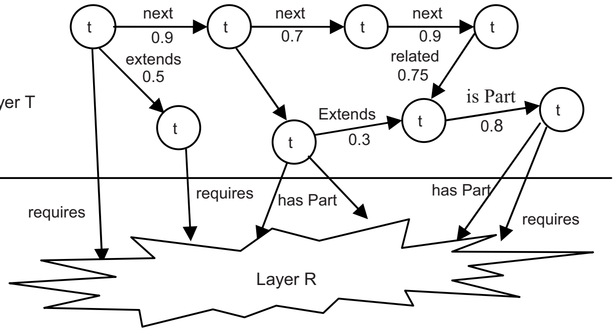


Figure 7. The relations graph, for T layer

2.2 Layer R – the frame library

The frame R contains the application part - fundamental unit:

- the frame identifier,
- evaluation methods,
- mutual relationship within layers (T and U).

The paragraph expresses the evaluation methods and applications features, defining their quality measures [1], [2], [3], [14], for single frames (being a static grade).

The defined solution uses single–argument operators ($^1\Phi$) as an evaluation tool, for separate components of the sequence **s** (**s** = <s_1, s_2, s_3, …, s_n>, where s_i – elementary answer-data). Products of interaction sequence are evaluated, by n-argument operators ($^n\Phi$). The n-argument operators allow evaluating mutual dependences between answers sequences. In Figure 8 the frames of the evaluation processes were presented.

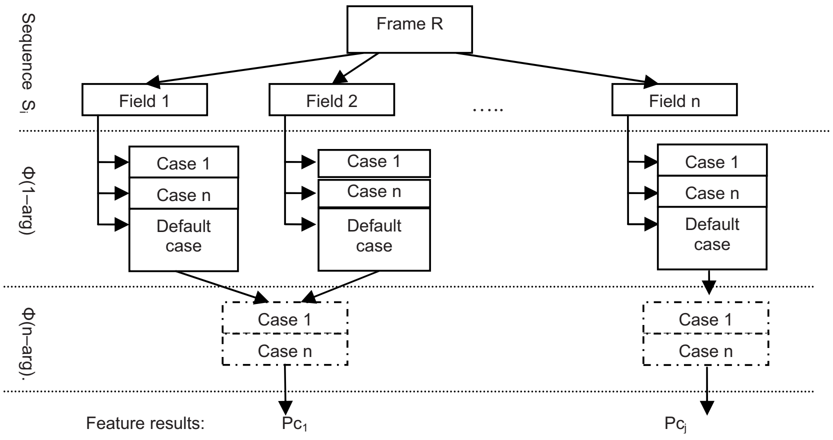


Figure 8. The MAMS evaluation process by 1Φ and nΦ operators

The used operators are based on the MAMS implementation engine functions [14], with the following operators:

- identity; for the returned values standardisation,
- comparison; for a measure equality definition, (eg. Knuth–Morris–Pratt [16]),
- extended, for user's pre-defined operators.

The classical solution gave us a single drawback only, considering every answer field s_i separately, as the independent one, distinguished from the other field's s_j for $j \neq i$:

$$\forall s_i, s_j, s_i \in S, s_j \in S, j \neq i \;\; P(s_i|s_j) = \frac{P(s_j \cap s_i)}{P(s_j)} = P(s_i) \qquad (13)$$

where:

P - is probability, defining mutual correlation of the results s_j and s_i.

The extensions into n-argument operators allow avoiding the inconvenient limits in the development process. The n-argument operator establishes mutual co-relations within the values of separate answers fields. The operator ${}^n\Phi$, provides several tools like logical conjunction (and, or), statistics, time measures, etc.

The implemented measures allow us evaluating the not exactly defined values. Moreover the new operators can be added into the system dynamically; any time it is needed. Values returned from the evaluation functions (F): $\{p_1, p_2, p_3, p_4, p_5, \ldots, p_n\}$, are taken under account while defining *overall grade features;* for the evaluation formula, as:

$$F_{\text{overall grade}} = \frac{\sum_{i=1}^{n} \lambda_i p_i}{\sum_{i=1}^{n} \lambda_i} \qquad (14)$$

The static evaluation measures, defined above, describe user's interaction within a single application frame. The dynamic evaluation procedures concern the frames sequences, based on a static set of the features (Figure 9).

The new dynamic feature values generation, for the frame r_n and preceding frame r_{n-1}, is defined by the following algorithm:

1. Any new c_i data for the frame r_n is available? if yes, fetch the new value p_n of the c_i, if not, stop the application,
2. Fetch the value p_{n-1} of the c_i, in the previous r_{n-1} frame (that gives us usually the default setting - $p_{n-1}=0$),
3. Find the difference:

$$\Delta\ p = |p_n - p_{n-1}|,$$

4. Find the function: $F_{^{d}c_i}(r_n, r_{n-1}) = (1-\Delta p)*\min(TU_S, WO_U)$ (15) and new value of the feature $^{d}c_{i.}$
5. Enter these new values into the application measures; E(r_{n-1}, r_n)= $F_{^{d}c_i}$,
6. Return to the step 1.

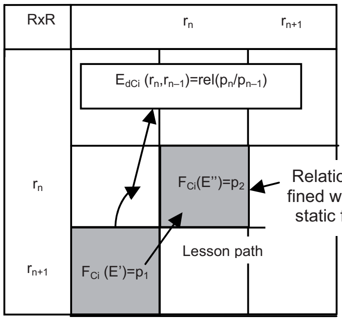

RxR
r_n
r_{n+1}
E_{dCi} (r_n,r_{n-1})=rel(p_n/p_{n-1})
r_n
F_{Ci}(E'')=p_2
Relation is defined within one static feature.
Lesson path
r_{n+1}
F_{Ci} (E')=p_1

Figure 9. The dynamic (on-line) relations within the layer R

The layer R defines the frame presentation mode as well as its evaluation features. The integration process of the R with T layers defines a key relation for the cognitive level definition.

2.3 The user's U layer

The users' layer (U) defines their personal data records; with their preference and abilities; introduced already in many investigations [1], [2], [3], [9]. The first two: the User's type (TU_s) and knowledge level (WO_u) define the main user's profile; using the description functions values, as:

$$F_{WO_s} \begin{cases} 0.99 & : \text{expert} \\ 0.6 & : \text{teacher} \\ 0.1 & : \text{student} \end{cases} \qquad (16),$$

The user's profile is mainly created on statistics, expressed by representative mean values, extracted as:

- the grade arithmetic mean value,
- the grade geometric mean value, illustrating the increasing user's knowledge,
- the dominant features, showing the most frequent user's grades,
- the quartile, first and third, for the grade distribution assignment.

Similarly, variety of additional measurements, as:

- variety domain, showing the results reliability,
- variances that show the average knowledge deviation finding,
- asymmetric and concentration measures, assigning the user's abilities in driving into lower or higher grades.

The function F, for statistical operation, is given by the operator:

$$E = U \times R$$
$$F : E \times C \to P_c \qquad (17)$$

The user's profile produces the valuable data for evaluation processes, with the following features: TU_S and OW_U, containing reliability values and verification measures.

The described structure creates unified repository of the e-content frames described by semi-natural words, enriched by the users' results. Structure is containing normalised values within features can be easily processed automatically. The developed processing solution is described next.

3. The application controlling unit

The interactivity engine concerns the lesson structure appropriate definition in accordance with the user's U layer-characteristic features. The conclusion making mechanisms implementation concerns term t_j and user u_i, illustrated in Figure 10:

$$\max(F_{c_{grade}} (t_j | u_i)) \tag{18}$$

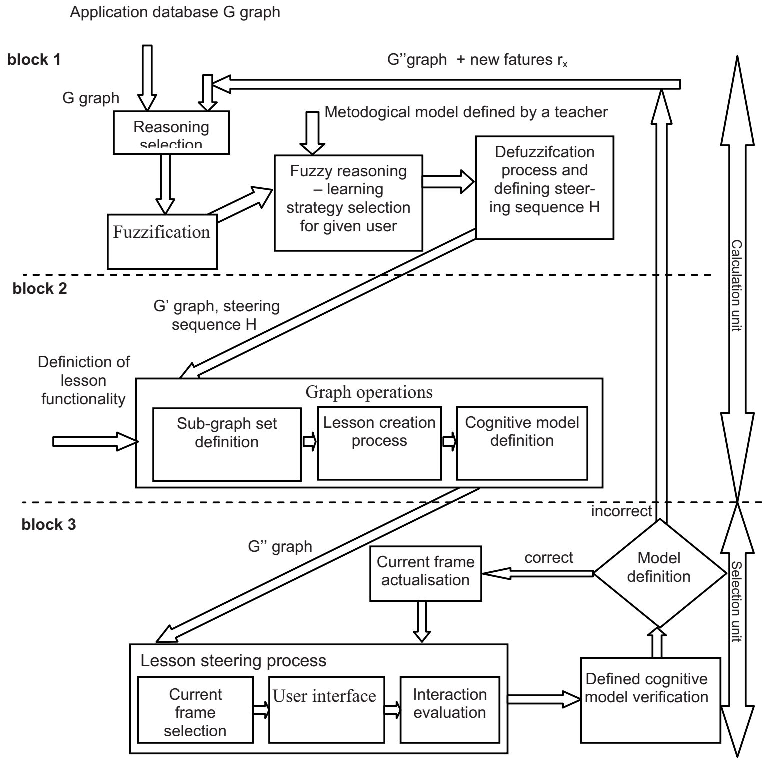

Application database G graph
block 1
G graph
G''graph + new fatures r$_x$
Reasoning selection
Metodogical model defined by a teacher
Fuzzy reasoning – learning strategy selection for given user
Defuzzifcation process and defining steer-ing sequence H
Fuzzification
block 2
G' graph, steering sequence H
Definiction of lesson functionality
Graph operations
Sub-graph set definition
Lesson creation process
Cognitive model definition
Calculation unit
block 3
incorrect
G'' graph
Current frame actualisation
correct
Model definition
Selection unit
Lesson steering process
Current frame selection
User interface
Interaction evaluation
Defined cognitive model verification

Figure 10. Block diagram of the lesson structure

Algorithm (block 1) defines the user's fuzzy reasoning procedures [22] that allow reducing the data set by filtering relations, placed between the users' and lessons' descriptors. The methodological model classifies the application features, modifying their weights (λ). Additionally the first block generates controlling sequence ($H=<h_1, h_2,\ldots,h_n>$), which is processed by the second block - the conclusions.

The conclusion making unit also defines the type of the graph-operations, needed for the lesson structure modifications, by:

- sum: $G_{sum} = G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$ (19),
- multiplication: $G_{multi} = G_1 \cap G_2 = (V_1 \cap V_2, E_1 \cap E_2)$ (20),
- α–cut, for all vertices, with at least one edge-value grater than
  α : $G_{cut} : v_i \in V', F(E_{V'xV''}) > \alpha)$ (21),
- sub-graph $(\cup C)$, defining the graph relation to the features: $G_{subgraph} : e \in E, F : E \times C \neq \phi)$ (22),
- the shortest path algorithm; based on Dijkstra theories [16] ,
- the extended path algorithm,
- the maximal flow algorithm, based on the Ford–Fulkerson algorithm [18].

The controlling system allows defining the lesson's repetition structure, adequate to the methodological model; of the lesson and courseware.

The complexity of the controlling sequences was assigned by the relation:

S=<OG, EL, H, FL>:

$$
\begin{aligned}
&OG = \{OG_1, OG_2, \ldots, OG_k\}\\
&EL = \{G'_1, G'_2, \ldots, G'_l\} : \bigcup_{i=1}^{l} G'_i \subseteq G'\\
&\forall\, el \in EL : el = (G'_1, G'_2, \ldots, G'_m) \quad \wedge \quad m \le l\\
&H = \{h : h = (h_1, h_2, \ldots, h_u) \wedge u \le g \wedge g \in N^+\}\\
&FL = (FL_1, FL_2, FL_3, FL_4)\\
&\forall\, i \in \{1,2,3,4\}\; FL_i = OG_{i_1} \circ OG_{i_2} \circ \ldots \circ OG_{i_w}, \quad \textit{where}\\
&\forall\, j \in \{1,\ldots, w\}\; OG_{i_j} \in OG\\
&\forall\, i \in \{1,\ldots, k\}\; OG_i : EL \to EL\\
&\forall\, OG_i, i \in \{1,\ldots, k\}\; \exists!\, h^j = (h'_1, h'_2, \ldots, h'_b),\; j \in \{1,\ldots, w\}\\
&\forall\, i \in \{1,\ldots, b\}\; \exists!\; j \in \{1,\ldots, u\}\; h'_i = h_j \wedge b \le u\\
&\forall\, i \in \{1,\ldots, w\}\; OG_i = OG_i(el) = OG_i(G^x_1, \ldots, G^x_m) \quad \textit{and} \quad \textit{where}\\
&\forall\, i \in \{1,\ldots, m\}\; \exists!\; j \in \{1,\ldots, l\}\; G^x_i = G'_j \wedge m \le l\\
&\bigcup_{i=1}^{\lg} G'_i \equiv G''
\end{aligned}
\tag{23}
$$

The XML representation:

```
<function name="repetition">
    <item typ="O" name="difference">
        <item typ="G">OR: repeat</item>
        <item typ="O" name="sum">
        <item typ="F">path</item>
        <item typ="O" name="succ">
        <item typ="F">path</item>
        <item typ="G">RO: grade</item>
    </item>
</function>
<function name="path">
….
</function>
```

where:

OG - is the set of the graph operator,
EL - concerns the sub graph of G’ graph,
H - assigns the controlling sequences,
FL - is the functions set, defining the lesson’s structures; a graph G’’ representations.

The G’’ graph, is a sub-graph of G’ graph. It contains only a lesson structure description, divided into the lesson’s tasks; assigned by functions of the MAMS shell:

$$FL_{path}, FL_{repetition}, FL_{evaluation}, FL_{verification} \tag{24}$$

If the presentation time exceeds maximal value (variable h_j) the aim t_i is treated as the courseware descriptors t_l for l = 1..m, defining the variable t_i, treated as the lesson’s aims:

$$\xi\mathrm{ti} = \bigcup\nolimits_{l=1}^{m} \Gamma_{t_l}, \tag{25}$$

The functions FL allow us to follow the lesson’s knowledge progress and to verify its measures (Figure 11). The algorithm organisation is defined by the graph structure.

The basic structure is defined as vertices and edges of the *path* function. The starting point in lesson is a free vertex – a first key descriptor (t_x).

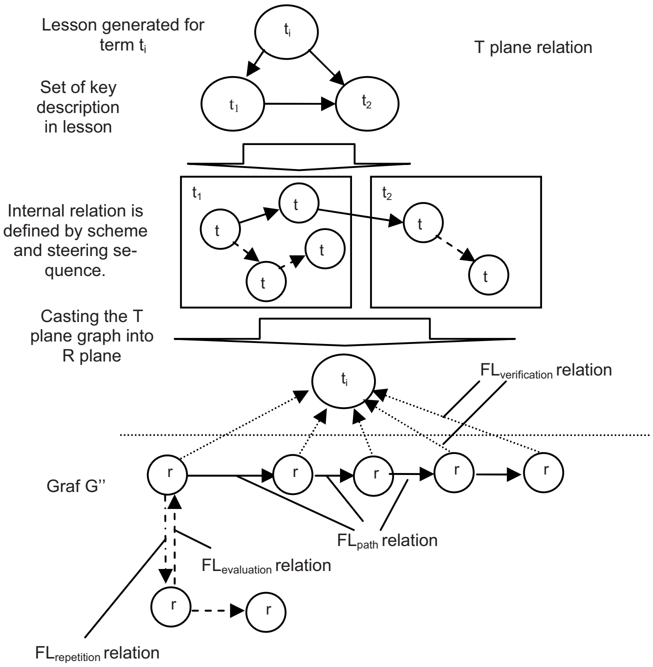

Lesson generated for term t_i
t_i
T plane relation
Set of key description in lesson
t_1
t_2
t_1
t_2
t
t
t
t
t
t
Internal relation is defined by scheme and steering sequence.
Casting the T plane graph into R plane
t_i
$FL_{verification}$ relation
Graf G’’
r
r
r
r
r
FL_{path} relation
$FL_{evaluation}$ relation
r
r
$FL_{repetition}$ relation

Figure 11. Block diagram of the lesson based on the G’’ graph

The application repetition number *lr* is measured by the complexity feature of the frame (p_r), with h_i characteristic values, as:

$$lr = \left\{ \begin{array}{l} 1 : p_r <= h_i \\ \frac{p_r}{h_i} : p_r > h_i \end{array} \right\} \qquad (26)$$

The selected frame returns the user's interactions into the R layer. The course flow follows the defined graph structure, according to the algorithm:

1. the user's interactions evaluation, produces the functionality $FL_{evaluation}$ features,
2. a next edge $E(r_x,r_i)$ from $FL_{repetition}$, is fetched, where r_x corresponds to the evaluated frame, r_i indicate the next available frame,
3. the value of *evaluate* function $E(r_x,r_i)$ is smaller than the value of the edge $E(r_i,r_x)$, defining the *repetition* function, then frame r_i is added into the candidate of the next frame; for the application path (ZK) definition,
4. if the unchecked edge exists, for vertices r_x, go to the second step of the application,
5. if ZK is empty, set the next frame r_i as it is appointed by the highest value of edges $E(r_x,r_i)$ for the *path* functions then exits the algorithm,
6. select the frame from ZK set, for the repetition edges that is multiplied by a value of its feature λ, indicating its weight.

4. Conclusions

The application database is provided by its unique relations into the recently defined courses; in contrary to the SCORM's and IMS's standards, where these descriptors are not modified automaticallyy (static structure).

The new courses can remain independent; where the courseware term set (T') is a unique one: (T''): $T' \cap T'' = \phi$ *and* $T' \cup T'' = T$.

The individual assignment of the system was supported by an algorithm modifying dynamically the courseware structure.

The algorithm adapts itself the expected training methodology with the application control preferences. The solutions available in SCORM and IMS are based on tree structures [4], [8] where the application structure freedom is far from flexibility.

The elaborated shell MAMS were also supported by the graphs operations and fuzzy measures implementation.

The expert analysis can easily modify the course structure, not only thanks to the database specific rules for the conclusions making, but also by a new model of the interactions measures recognition.

In Figure 12 the advantages of the introduced model (part b) are visible, where theoretical and practical aspects of the graph implementation were discussed [9], [10], [18], [25] (part of the Figure 12).

The multi-functions graph descriptors (of the MAMS four graphs: *path*, *grade*, *repetition* and *verification*) are intuitive and can be easily be verified by an expert. Moreover the graphs descriptors can be converted into other standards, allowing create a unified platform for the future courses definitions.

The automatic conclusion making system, using many advanced quality measures, was roughly discussed in this contribution. The elaborated solutions unify various tools for e-learning unit's development.

The user's data record, provided by the application terms, describe the user's knowledge in the application area. Thanks to the graph theories implementation the application structures description can be modified continuously, collecting all previously defined evaluation bases.

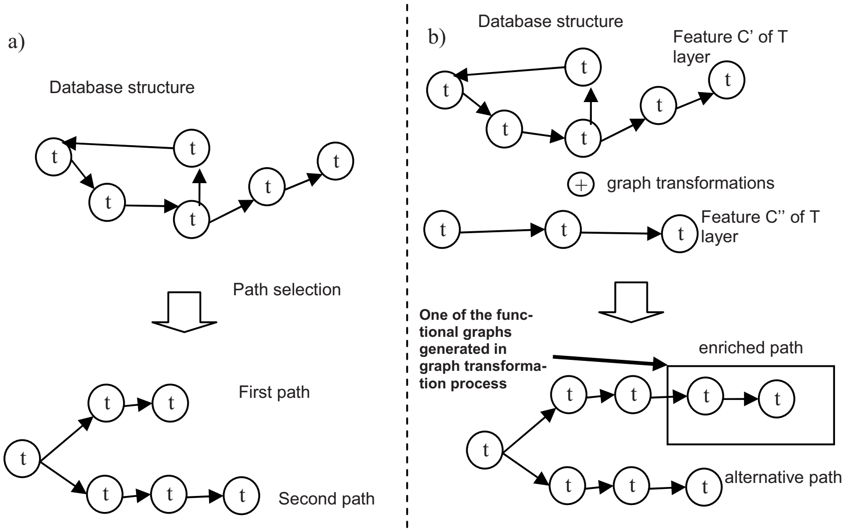


Figure 12. The path creation result a) available solution [9], [10], [18], [25], b) proposed solution

The introduced solution offers the data system, providing the user with a form of intelligent tutoring system that is controlled by a full range of functions, used for driving flexible the learning management platform.

The results and methods, using variety of possible interactions, can still be modified, in case any additional functions are needed. Further evaluation works are still in progress, with relationship analysis to many other works [23], [24], [25].

References

[1] Piecha, J. The multi-level model for microcomputer ICAI system. *The International Journal of Applied Engineering Education*, Vol. 5, No 3. pp.391-395, Pergamon Press: Oxford, NewYork, Tokyo, 1989.

[2] Piecha, J. The programmable shell for multimedia applications development. *Journal of Applied Computer Science*, Łódź 1999.

[3] Piecha, J. Data units selection within sequential database. *Journal of Medical Informatics and Technologies*, Vol.2/2001, ISSN 1642-6037, pp. MI 155-162.

[4] Specification: *SCORM 2004 3rd Edition Overview*. Advanced Distributed Learning. http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/2004%204th %20Edition/Documentation.aspx, downloaded: September, 26th 2008.

[5] Specification: *SCORM 2004 3rd Edition Content Aggregation Model (CAM) Version 1.0*. Advanced Distributed Learning. http://www.adlnet.gov/Technologies/scorm/SCORMSDocuments/2004%204th%20 Edition/Documentation.aspx, downloaded: September, 26th 2008.

[6] Specification: *SCORM 2004 3rd Edition Run-Time Environment (RTE) Version 1.0.* Advanced Distributed Learning. http://www.adlnet.gov/Technologies/scorm/ SCORMSDocuments/2004%204th%20Edition/Documentation.aspx, downloaded: September, 26th 2008.

[7] Specification: *IMS Learning Design Best Practice Guide.* http://www.imsglobal.org/learningdesign/ldv1p0/imsld_bestv1p0.html, downloaded: January, 5th 2009.

[8] Specification: *IMS Learning Design Information Model.* http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html, downloaded: January, 5th 2009.

[9] Piecha, J. Data units selection within sequential database*. Journal of Medical Informatics and Technologies.* Vol.2/2001, ISSN 1642-6037, pp. MI 155-162.

[10] Technical specification of Topclass LMS, http://www.wbtsystems.com/news/release/1291, downloaded: February, 17th 2009.

[11] Technical specification of Lotus Leasing Space, http://www306.ibm.com/software/lotus/, downloaded: February, 17th 2009.

[12] WWW page of LMS: Total LMS, LCMS, http://www.sumtotalsystems.com/products/datasheets/lcms.pdf, downloaded: February, 17th 2009.

[13] Technical specification of Moodle platform. http://docs.moodle.org/en/CVS_%28 developer%29, downloaded: February, 17th 2009.

[14] Piecha, J.; Król, R.; Olszewski A. The multimedia files development in frames of interactive fast explorer*. Journal of Medical Informatics and Technologies*, Vol.5/2003, ISSN 1642-6037, pp. MI – 155-163

[15] WWW Webpage of Eurydice portal of Erasmus program http://www.eurydice.org, downloaded: January, 5th 2009.

[16] Bałachowski, L.; Diks, K.; Rytter W. *Algorytmy i struktury danych*. Wydawnictwa Naukowo - Techniczne, 2006.

[17] Motwani, R.; Raghavan P. *Randomized Algorithms.* Cambridge University Press, 1995.

[18] Rutkowski L. *Metody i techniki sztucznej inteligencji.* Wydawnictwo Naukowe ISBN: 83-01-14529-3, PWN 2006.

[19] Ogiela, L.; Tadeusiewicz, R.; Ogiela, M. R. A New Approach to the Computer Support of Strategic Decision Making in Enterprises by Means of a New Class of Understanding Based Management Support Systems*. 6th International Conference on Computer Information Systems and Industrial Management Applications* (CISIM'07) 2007.

[20] Forbus, K.D.; Hinrisch, T. R. Companion congnitive systems: A step toward human-level ai*. AI Magazine*, 27: pp. 83–95, 2006.

[21] Forbus, K.D.; Hinrisch, T. R. Engines of the brain: The computational instruction set of human cognition*. AI Magazine*, 27: pp. 15–31, 2006.

[22] Nguyen, T.T.; Skowron, A. Rough set approach to domain knowledge approximation*. Proceedings of the 9-th International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing* (RSFDGrC'2003), Chongqing, China, Oct 19-22, Volume 2639 of LNCS, pages 221–228. Springer, Heidelberg 2003.

[23] Doina, B.; Bich-Lien. D.; Yolaine, B. A Semi-Automatic Tool for Persnalizing Reusing Web Learning Resources. *Seventh IEEE International Conference on AdvanceLearning Technologies*, 2007.

[24] Feng-Hsu, L.; Wang, A. Contextualization Method of Browsing Evets in Web-based Learning Content Management System for Personalized Learning. *Seventh IEEE International Conference on AdvanceLearning Technologies*, 2007.

[25] ChaoBoon, T.; Kheng Leng, G. A Knowleadge-Driven Model to Personalize E-Learning. *ACM Joural of Education Resources in Computing*, Vol 6, No 1, article 3 March 2006.