Interactions Validation Methods for Training Resources Control Engine Development

Marcin Bernaś, Jan Piecha
Informatics Systems Department of Transport
Faculty of Transport, Silesian University of Technology
ul. Krasińkiego 8, 40-019 Katowice, Poland
marcin.bernas@gmail.com, jan.piecha@polsl.pl

Abstract. *The courseware complexity selection is one of the most difficult factors, of an intelligent application engine development. An individual application starting level selection allows setting the most adequate application structure, to the users' expectations. What is more this structure allows dynamically control the application in time it is used.*
The application description units and their controlling techniques are bounding the data base components into individual composition of the course.
The paper introduces various aspects of distance learning resources development, fulfilling these demanding assumptions.

Keywords. LMS, ITS, e-learning, MAMS.

1. Introduction

The electronic courses structure and organisation were defined by various specifications and they are controlled by Learning Management Systems (LMS), in frames of these theories and standard unifications [1][4][6][7]. The training applications (e-content) are developed in fundamentals of various theories [3][8] based on different forms: linear paths, branched trees, blocks and sequences, complex graphs.

The training courseware consists of fundamental units called frames or Sharable Content Objects (the smallest organisation data-unit) [5]. The application current path selection proceeds in accordance with the scheme, introduced by block diagram in Fig. 1.

The user interface distinguishes the starting values for an application [9] and further interactions for the application control. Alternative solution uses several introductory questions (pretests) that set the data of the courseware; according to the user knowledge level [9].

Majority of the LMS units [10][11][12][13] estimate the knowledge by simple measures expressed in percentage form 0% to 100% from single or multiple choice answers.

For simple evidences the single or multiple choice and single fill-up format are satisfying for electronic training systems users. However one can find many examples where these data-recording models are not satisfying at all. The presentation frame expresses more sophisticated content where traditional interactions are not effective enough for defining the user knowledge. Moreover several evaluation algorithms are implemented for the user's interactions analysis.

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

Calculation assigns the application unit working for evaluation of the user's interactions feedback. The obtained results fulfil the assumed requirements for presentation content and for the application repetition [14].

The application frame mode is set to one – for still active frames and to zero for lessons excluded for current presentation. The frames mode is set in the *selection module.*

The paper is dealing with reliability of evaluation methods with multiple classes of the answers classification; used as a quality control algorithm for new solutions. The block diagram presented in Fig. 3 introduces a proposal for more complex evaluation procedure of user's interactions. The extended structure of the evaluation algorithm (assigned by A) uses the knowledge database content for the classification criterions development.

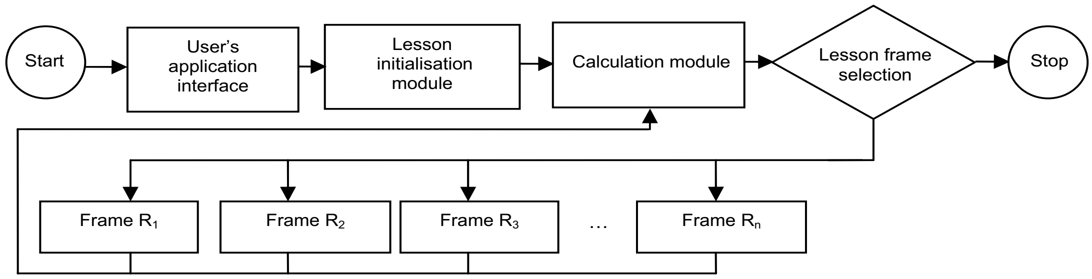

Figure 1. Block diagram of the lesson path selection

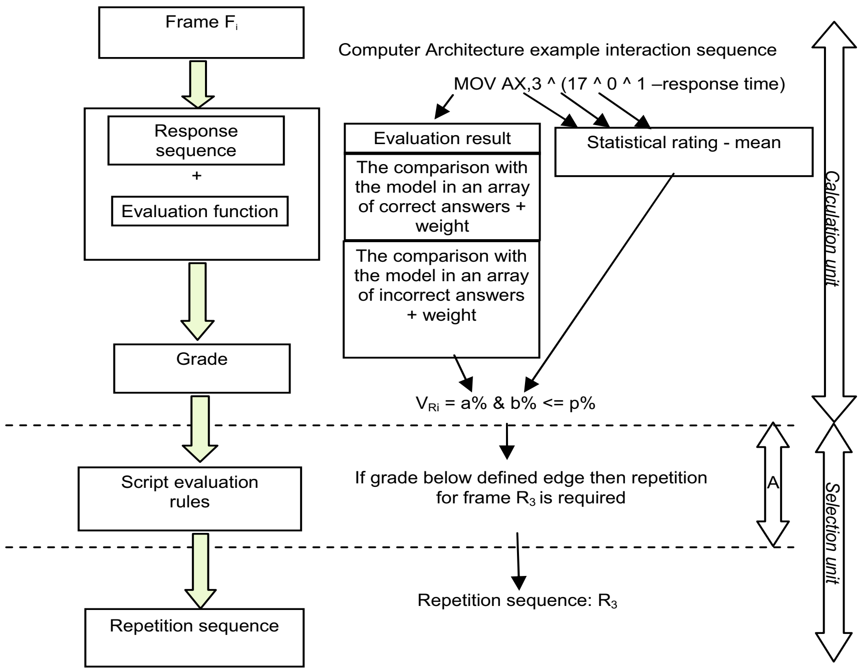

Figure 2. Block diagram of lesson current path selection algorithm

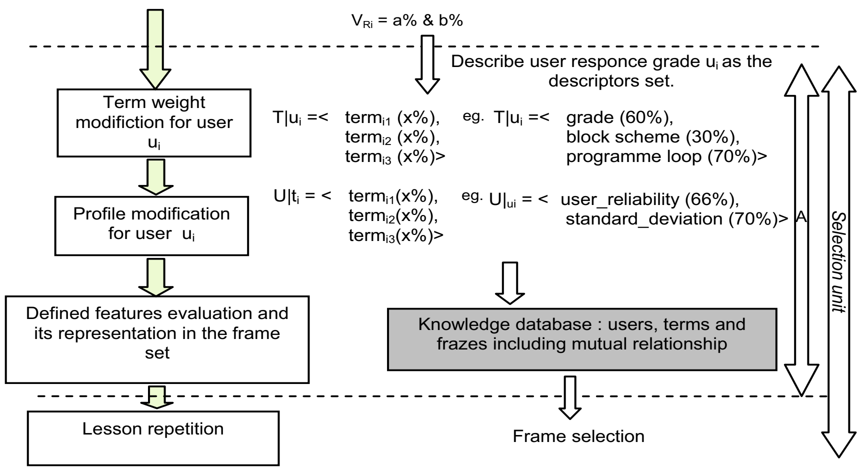

Figure 3. The application control flowchart through the knowledge database

The control unit was applied in Multimedia Applications Management Shell (MAMS) in its control layer [2][3].

The application controlling data assigns personalisation characteristic features. The lesson structure is constructed on the knowledge data record. The algorithm discussed in this paper modifies the application controlling processes, including additional personal abilities of the user. The lesson is constructed by data from the user script interface supported by weight modifiers added to the data units. They assign current knowledge level of the user.

The knowledge database structure is divided into the objects classes:

- Users; with their profiles (abilities and skills),
- E-content; with the frames identifiers,
- Terms; defining knowledge descriptors.

Two types of the classes are available in majority of learning process controllers (management systems - LMS). The paper discusses the characteristic features of the classes, with additional descriptors (Fig. 3), complementing the so called Intelligent Tutoring Systems (ITS). This way an automatic tutoring machine was offered.

The value of feature relation with object is described by F function, which distinct mutual relation of database elements within one or two layers. The integrity of structure is achieved by relation shown on fig. 4. The above relations require graphs assignments that allows define links between data single units (frames) and more complex structures of the applications, as lessons and courses.

The graph description gives more design freedom to course composition and complements the tree structures at the same time (used for e.g. in SCORM - well known standard applications [4]). The solution contains semantic description, available in majority of thesauruses and syllabuses; based on RDF [26] (*Resource Description Framework*) and TESE [15] (*Thesaurus of European Search Engine*) none similar relations.

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

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

where:

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

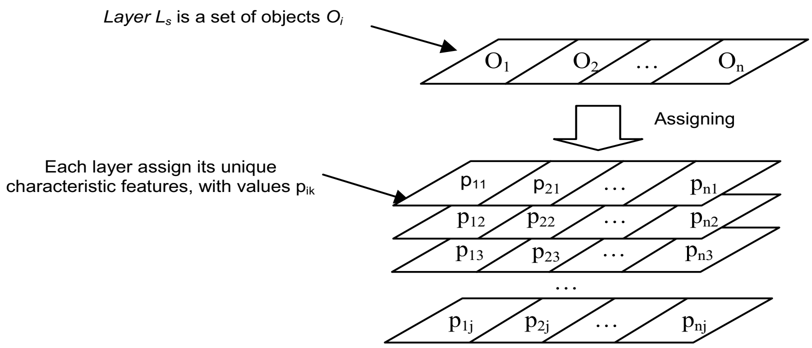

Figure 3. Graphical layers definition.

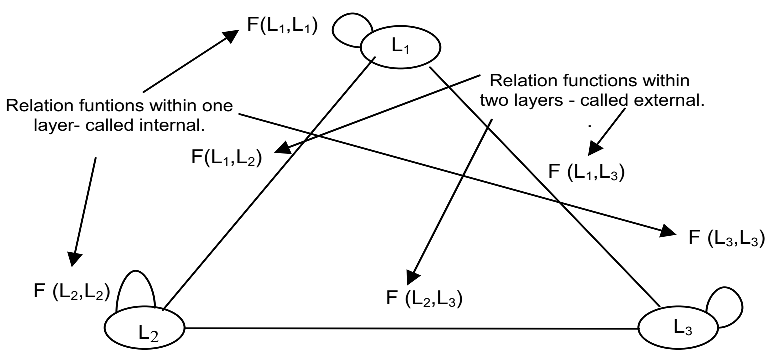

Figure 4. References between layers.

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

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

$$V = R \cup T \cup U \quad (2)$$

$$R \cap T = R \cap U = T \cap U = \phi \quad (3)$$

where:

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

$R = \{r_1, r_2, \ldots, r_k\}$

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

$T = \{t_1, t_2, \ldots, t_m\}$

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

$U = \{u_1, u_2, \ldots, u_n\}$

The layers R, T, U of the graph:

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

where:

- v_i - the object (vertices) in the L layer,
- L - distinct layer of objects and features,
- C_L - the L layer of finite set of attributes,
- P_{c_L} - the set of layers attributes values,
- F^V - the default function, assigning the value p_{cj} of objects v_i and feature c_j,
- λ - quality function, influence on evaluation and lesson selection process (default = 1) of specified feature.

Each layer is described by number of standardised features, where its values are defined as:

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

or

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

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).

$$\forall c_{Lj} \in C_L \;, F^V \;: V \times C \to \{p_C : p_C \in [0,1]\} \quad (7)$$

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

$$G : e_i \Rightarrow V \times V \quad (8)$$

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

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

where:

- $E_{L'\times L''}$ -edge for an ordered pair of vertices,
- F^E -default function, assigning value p_{cj} for edge e_i and feature c_j,
- $C_{L'xL''}$ -finite set of features (attributes) for Cartesian product (layers L'xL''),
- $P_{C_{L'xL''}}$ -values set for distinct attributes. Values are within the range [0,1],
- λ -quality function, defines evaluation results for the lesson selection (default=1).

Both functions, F^V and F^E are the Cartesian products. Equation 6 expresses the functions generalisation procedure:

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

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 vertice. The range ([0,1]) of the function values, simplifies the fuzzy understanding conclusions algorithms.

2. The characteristic features definition with their functions

2.1 The layer T description in semi-natural language

The T layer features were defined by in this works; for RDF standard [26] implementation [20][21]. Terms layer (T) was defined using 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,
- function it is a pre-defined word type, in the specified language,

- abstraction level is based interpreted by the author's grade,
- description contains the information unit expressing the type, content or additional description units (not evaluated) - $F_{descrition}=0$.

The T features are words, with 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 [26][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:

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

and

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

where:

$C|E$ -the given feature edge,

p -weight of the relation defined by the thesaurus.

T layer allows us defining the key idea of the lesson. Next step is connections finding of layers with application's frames (layer R): Possesses the application Part, Requires the unit, Broader, Narrower.

The T layer defines the fundamental structure of the application. The example layer was introduced in Fig. 5.

2.2 Layer R – the frame library

The frame R contains the application part - fundamental unit:
- the frame identifier,
- evaluation methods,
- linking the application into one structure (layers T and U).

The paper is concentrated onto evaluation methods and features, defining quality measures. [1][2][3][14].

Evaluation methods that concern the operator functions (Φ):

$$\begin{array}{l}
{}^1\Phi = ({}^1\Phi_1, {}^1\Phi_2, \ldots, {}^1\Phi_l) \\
{}^n\Phi = ({}^n\Phi_1, {}^n\Phi_2, \ldots, {}^n\Phi_m) \\
\forall s \in S : s = (s_1, s_2, \ldots, s_l) \\
\forall b \in B : b = (b_1, b_2, \ldots, b_u) \\
P = \{p_1, p_2, \ldots, p_w\} \\
F_{c_j} = {}^n\Phi \circ ({}^1\Phi \circ {}^1\Phi \circ \ldots \circ {}^1\Phi), where: \\
\forall i \in \{1,\ldots,k\} \;\; {}^1\Phi_i : S \times B \rightarrow P \\
\forall i \in \{1,\ldots,k\} \;\; {}^1\Phi_i = {}^1\Phi_i(s,b) = {}^1\Phi_i(s_i, b'_1, \ldots, b'_t) \;\; and \;\; where: \\
\forall i \in \{1,\ldots,t\} \;\; \exists! \;\; j \in \{1,\ldots,n\} \;\; b'_i = b_j \wedge t \le n \\
\forall i \in \{1,\ldots,k\} \;\; {}^n\Phi_i : P \times P \times \ldots \times P \times B \rightarrow P_c \\
\forall i \in \{1,\ldots,m\} \;\; {}^n\Phi_i = {}^n\Phi_i(p'_1, \ldots, p'_z, b'_1, \ldots, b'_r), where: \\
\forall i \in \{1,\ldots,z\} \;\; \exists! \;\; j \in \{1,\ldots,w\} : p'_i = p_j \wedge z \le w \\
\forall i \in \{1,\ldots,r\} \;\; \exists! \;\; j \in \{1,\ldots,n\} \;\; b'_i = b_j \wedge r \le u \\
z + r \le n
\end{array} \quad (13)$$

where:

c_j - evaluation of j feature,

${}^1\Phi$, ${}^n\Phi$ - operator function family: one- or multi–argument.

s - sequence for given feature,

B - values sequence for operator function,

P - ${}^1\Phi$, ${}^n\Phi$ function range.

Defined solution uses one–argument operators (${}^1\Phi$) as evaluation tool for separate element of sequence **s**. Products of interaction sequence is evaluated using n-argument operators (${}^n\Phi$). N-argument operators evaluate mutual dependences between answer sequences. Fig. 6 presents frames evaluation process.

The user answers (**s**) are returned into the evaluation unit – strings unit analysis. However the binary data representation (by graphic or voice modulation) is also accepted. The data sequence **s** belongs to the sequence:

$$\mathbf{s} = \langle s_1, s_2, s_3, \ldots, s_n \rangle,$$

where variables of the sequences (fields) **s** are separated by s_s separation mark. Every field s_i is defined as:

$$s_i = \langle z_1\ z_2\ \ldots\ z_i \ldots \rangle$$

where:

z_i - any character or number in a text form. If binary value is returned, it is preceded by s_z separation mark. For binary data s_i is given by syntaxes:

$$s_i = \langle\ s_z z_1\ z_2\ \ldots\ z_i \ldots \rangle$$

The answer sequence is evaluated by composition of the evaluation functions.

The MAMS + QRU[27] use one-argument operators, by the relation:

$${}^1\Phi_j : S \times B \rightarrow P' \quad (14)$$

where:

B - values of evaluation parameters,

S - a value of answers sequence,

P' - classification results.

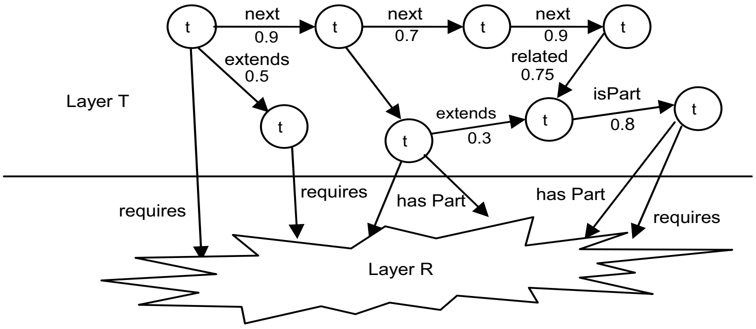

Figure 5. The relations graph, for T layer

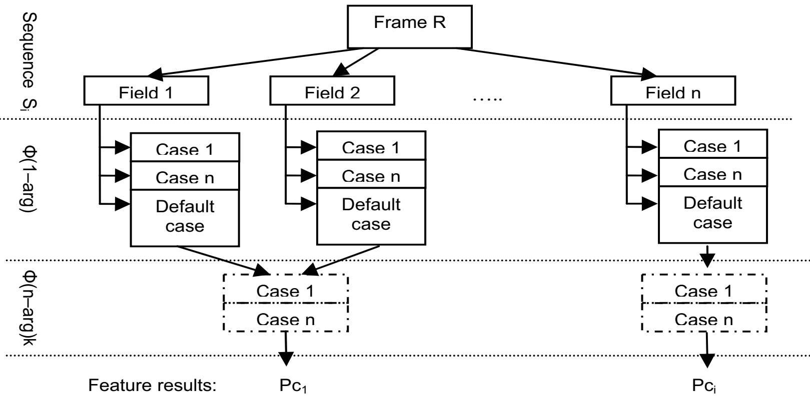

Figure 6. The MAMS evaluation process by $^1\Phi$ and $^n\Phi$ operators

Operators are based on the MAMS implementation engine functions [14], with operators:
- identity; for returned values standardisation,
- compare; for measure equality definition, (Knuth–Morris–Pratt [16]),
- default, for standard evaluation value.

The classical solution gave us one drawback only. It considered every answer field s_i as the independent one, from other field's s_j where $j \neq i$:

$$\forall s_i, s_j, s_i \in S, s_j \in S, j \neq i \;\; P(s_i|s_j) = \frac{P(s_j \cap s_i)}{P(s_j)} = P(s_i) \qquad (15)$$

where:

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

The author's improvement of the MAMS platforms for n-argument operators rejected these restriction limits. The n-argument operator establishes mutual co-relations within the values of separate answers fields (operator based on equation 13), as:

$$^n\Phi : P' \times P' \times \ldots \times P' \times B \rightarrow Pc_j \qquad (16)$$

It is possible to overpass the one-argument operator evaluation processes by using $^1\Phi_\wedge$ operator. The operator $^n\Phi$, provides several tools: logical conjunction (and, or), statistics, time measures, etc.

$$^n\Phi \; : S \times B \rightarrow Pc_j \qquad (17)$$

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

$$F_{overall \;\; grade} = \frac{\sum_{i=1}^{n} \lambda_i p_i}{\sum_{i=1}^{n} \lambda_i} \qquad (18)$$

The static measures, defined above, describe interaction process within single frame. The dynamic evaluation features define data sequences based on a static set of the features; ^dc_i is based on r_n and r_{n-1} frame c_i feature value (Fig. 7).

The dynamic features values calculations of the frame r_n:

1. any new c_i data for the frame r_n available? – if not, stop the application, if it exist, fetch new value p_n of the c_i,
2. fetch the value p_{n-1} of the c_i, of the previous r_{n-1} frame (default setting: $p_{n-1}=0$),
3. find the difference:

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

4. find the function:
$$F_{{}^{d}c_i}(r_n, r_{n-1}) = (1-\Delta p) * \min(TU_S, WO_U) \quad (19),$$
new value for the feature ${}^{d}c_i$.
5. enter these new relations into the application structure; $E(r_{n-1}, r_n) = F_{{}^{d}c_i}$,
6. return to the step 1.

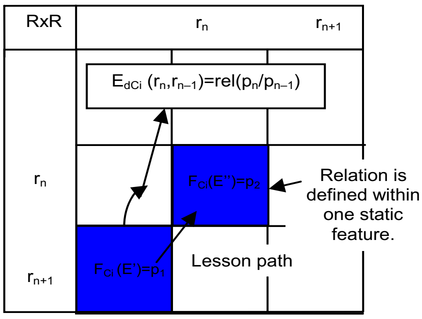

Figure 7. Dynamic relation within R

The interactions results – given by functions and assigned to U and R layers, defined for features $c_j \in C$:

$$\forall c_j \in C \; F_{user}(u_i, c_j) \equiv F_{c_j}(s|_{u_i}) \quad (20)$$

The layer R defines the frame's presentation unit as well as the evaluation features.

Integration of the R with T layers defines key relation for the cognitive result definition.

2.3 The user's U layer

The users layer (U) defines the data record; with its preference and abilities; based on big number of the research works [1][2][3][9]. First two: the User type (TU_s) and knowledge level (WO_u) defines the main user's profile; by the function values:

$$F_{WO_S}\begin{cases}0.99 : \text{expert}\\ 0.6 : \text{teacher}\\ 0.1 : \text{student}\end{cases} \quad (21),$$

The profile features are generated based on statistics. The most representative mean values were extracted from the system:
- grade arithmetic mean,
- grade geometric mean, illustration of increasing user's knowledge,
- dominant feature, showing the most frequent user grade,
- quartile, first and third, for the grade distribution

As well as variety of additional measurements:
- variety domain, shows the results reliability,
- variance that shows the average knowledge deviation,
- asymmetric and concentration measures, assigning the user's abilities to drive into lower or higher grades.

Function F for statistical operation is given by the operator:

$$E = U \times R$$
$$F : E \times C \rightarrow P_c \quad (22)$$

The user's profile is valuable data for the evaluation; features TU_S and OW_U contain reliability values –verification measures.

3. The application controlling unit

Next step of the interactive engine development concerns the lesson structure appropriate definition in accordance with the user's U layer-characteristics. The implementation concerns term t_j and user u_i (Fig. 8):

$$\max(F_{c_{grade}}(t_j|u_i)) \quad (23)$$

The learning strategy definition

The features for the user's fuzzifying reasoning procedure [22], reduce the extend set of the data performed by filters cutting relations between users' and lessons. Methodological model classifies the features modifying their weights (λ). The application controlling sequence ($H=\langle h_1, h_2,\ldots,h_n\rangle$) are processed by a second block

The lesson structure definition

The unit defines types of graph-operations need for the lesson structure definition:
- sum: $G_{sum} = G_1 \cup G_2 = (V_1 \cup V_2, E_1 \cup E_2)$ (24),
- multiplication: $G_{multi} = G_1 \cap G_2 = (V_1 \cap V_2, E_1 \cap E_2)$ (25),
- α–cut, for all vertices, at least one edge value is grater than α : $G_{cut} : v_i \in V', F(E_{V'xV''}) > \alpha)$ (26),
- sub-graph $(\cup C)$, the graph relation of the defined feature: $G_{subgraph} : e \in E, F : E \times C \neq \phi)$ (27),
- shortest path algorithm; based on Dijkstra [16] ,
- extended path algorithm,
- maximal flow algorithm, based on Ford–Fulkerson algorithm [18].

The system allows using the lesson functionality definition. The schemes were defined, based on methodology for lesson and course.

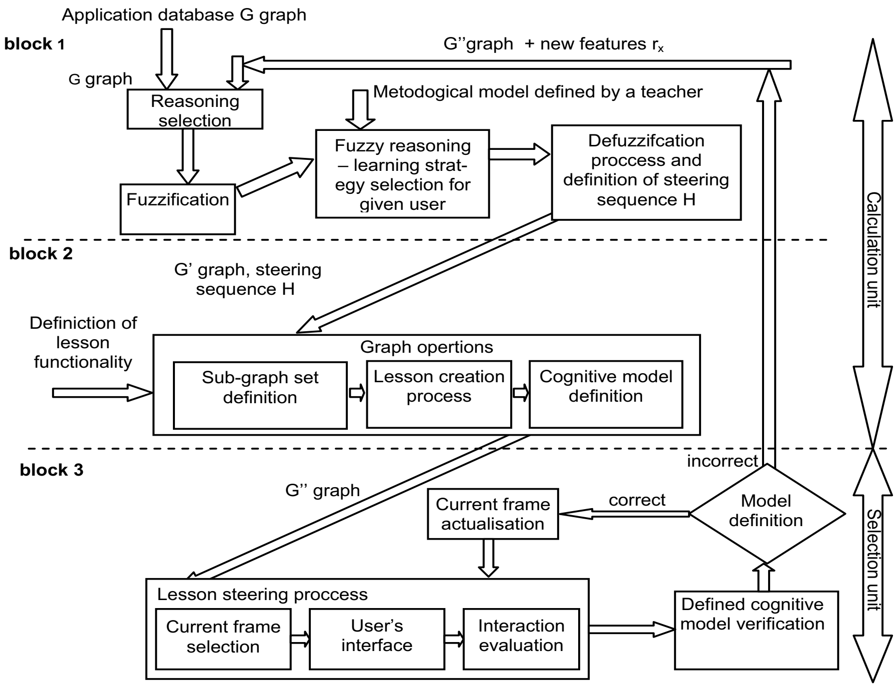

Figure 8 Block diagram of lesson algorithm

The complexity and controlling sequence were defined by: S=<OG, EL, H, FL>:

$$
\begin{aligned}
& OG = \{OG_1, OG_2, ..., OG_k\} \\
& EL = \{G_1^{'}, G_2^{'}, ..., G_l^{'}\} : \bigcup_{i=1}^{l} G_i^{'} \subseteq G' \\
& \forall el \in EL : el = (G_1^{'}, G_2^{'}, ..., G_m^{'}) \quad \wedge \quad m \le l \\
& H = \{h : h = (h_1, h_2, ..., h_u) \wedge u \le g \wedge g \in N^{+}\} \\
& FL = (FL_1, FL_2, FL_3, FL_4) \\
& \forall i \in \{1,2,3,4\}\ FL_i = OG_{i_1} \circ OG_{i_2} \circ ... \circ OG_{i_w}, \quad where \\
& \forall j \in \{1,...,w\}\ OG_{i_j} \in OG \\
& \forall i \in \{1,...,k\}\ \ OG_i : EL \to EL \\
& \forall OG_i\ , i \in \{1,...,k\}\ \ \exists! h^j = (h_1^{'}, h_2^{'}, ..., h_b^{'}),\ j \in \{1,...,w\} \\
& \forall i \in \{1,...,b\}\ \ \exists!\ \ j \in \{1,...,u\}\ \ h_i^{'} = h_j \wedge b \le u \\
& \forall i \in \{1,...,w\}\ \ OG_i = OG_i(el) = OG_i(G_1^{x}, ..., G_m^{x})\ \ and\ where: \\
& \forall i \in \{1,...,m\}\ \ \exists!\ \ j \in \{1,...,l\}\ \ G_i^{x} = G_j^{'}\ \ \wedge m \le l \\
& \bigcup_{i=1}^{\lg} G_i^{'} \equiv G''
\end{aligned}
\tag{28}
$$

where:

OG - set of graph operator,
EL - sub graph of G' graph,
H - controlling sequence,
FL - function set, defining the lesson structures; they are a graph G'' representations.

The lesson presentation and verification

The G'' graph is a sub-graph of G' graph. It contains only a lesson structure, divided into the lesson tasks; defined by functions:

$$FL_{path}, FL_{repetition}, FL_{evaluation}, FL_{verification} \tag{29}$$

If the presentation time exceeds maximal value, of variable h_j, the aim t_i is treated as course(ξ) and key descriptors t_l for l=1..m, defining t_i treated as lesson's (Γ) aims:

$$\xi_{ti} = \bigcup_{l=1}^{m} \Gamma_{t_l}\ , \tag{30}$$

Functions FL are used to follow the lesson progress and to verification its purposes (Fig. 9). The algorithm organisation is performed by the graph structure.

Basic structure is defined as vertices and edges of the *path* function. Starting point in lesson is a free vertex – a first key descriptor (t_x). Repetition number *lr* is measured by the complexity feature of the frame (p_r) with h_i characteristic values, as:

$$lr = \begin{cases} 1 & : \ p_r <= h_i \\ \dfrac{p_r}{h_i} & : \ p_r > h_i \wedge h_i \neq 0 \\ \infty & : \ h_i = 0 \end{cases} \tag{31}$$

The selected frame returns the user interactions into the R layer. If the obtained results are not matching the cognitive definition, algorithm recalculate the complex level of the application [19].

If the assumed requirements are fulfilled, the next frame of lesson is selected, according to the following algorithm:

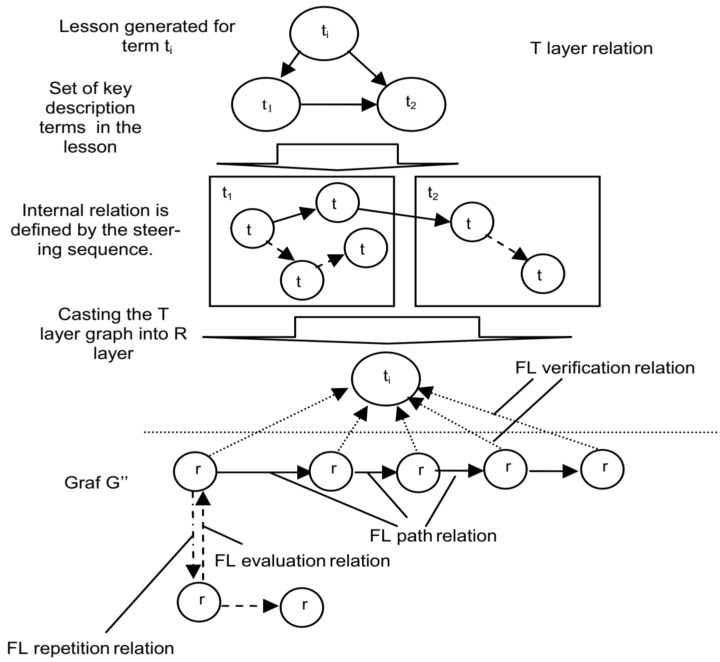

Figure 9. Block diagram of lesson generated based on G'' graph

1. evaluation of user interactions, analysing the functionality: $FL_{evaluation}$ features,
2. fetch a next edge $E(r_x,r_i)$ from $FL_{repetition}$, were r_x is currently evaluated frame, r_i is a next frame available,
3. if the value of *evaluate* function $E(r_x,r_i)$ is smaller than the value of edge $E(r_i,r_x)$, defining the *repetition* function, then frame r_i is added into the candidate for next frame set (ZK), in the application path,
4. if exists the uncheck edge, for vertices r_x, go to the second step of the application,
5. if ZK is empty, set the next frame r_i as appointed by the highest value of edges $E(r_x,r_i)$ for *path* functions and end the algorithm, select frame form ZK set, which value of the repetition edges is multiplied by its feature λ of the weight value.

4. Conclusions

The automatic conclusion system, using high quality measures, was roughly discussed. The solution unifies various features of the application database of database e-content development.

The user data layers, together with terms define the user's knowledge. Thanks to the graphs data descriptions the application structures can be driven continuously, with the same evaluation bases. The elaborated solution offers a hybrid data system, working in a form of ITS (intelligent tutoring system) with full range of functions available in LMSes (learning management systems). The results and variety of possible implementation are still under careful analysis in context to similar works [23][24][25].

References

[1] Piecha J. The multilevel 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. 2006 http://ADLnet.gov

[5] **Specification**: SCORM 2004 3rd Edition Content Aggregation Model (CAM) Version 1.0 , Advanced Distributed Learning. 2006 http://ADLnet.gov

[6] **Specification**: SCORM 2004 3rd Edition Run-Time Environment (RTE) Version 1.0. Advanced Distributed Learning. 2006 http://ADLnet.gov

[7] **Specification**: IMS Learning Design Best Practice Guide. IMS Global Learning Consortium 2003 http://www.imsglobal.org/learningdesign/ldv1p0/imsld_bestv1p0.html

[8] **Specification**: IMS Learning Design Information Model. IMS Global Learning Consortium 2003 http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html

[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

[11] **Technical specification** of Lotus Leasing Space http://www306.ibm.com/software/lotus/

[12] **WWW page** of LMS: TotalLMS,LCMS http://www.sumtotalsystems.com/products/datasheets/lcms.pdf.

[13] **Technical specification** of Moodle platform http://docs.moodle.org/en/CVS_%28developer%29

[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] W**WW Webpage** of Eurydice portal of Erasmus program http://www.eurydice.org

[16] Banachowski 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 Babu, Bich-Lien Doan, Yolaine Bourda A Semi-Automatic Tool for PersnalizingReusing Web Learning Resources, Seventh **IEEE International Conference** on AdvanceLearning Technologies, 2007

[24] Feng-Hsu 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 Teo, Robert Kheng Leng Gay. A Knowleadge-Driven Model to Personalize E-Learning. **ACM Joural of Education Resources** in Computing, Vol 6, No 1, article 3 March 2006

[26] **Resource Description Framework (RDF)** Model and Syntax Specification: www.w3.org/TR/REC-rdf-syntax/

[27] Król R., Bernaś M., Piecha J. The Mams Controlling Via Interactivity Measures. **Research Reports on e-Content and Distribution Platforms Technologies**. ISBN 83-922374-1-2. Katowice 2006, pp. 45-52