Generalized Net Model of a Program System for Composing Verses

Dilyana Budakova, Lyudmil Dakovski

***Abstract**. The article reviews a generalized net model of a program system for constructing verbal expressions and especially for composing verses. The process is decomposed in four subsequent stages of transforming an emotionally evaluated input set of key notions into corresponding, by sense and evaluations, grammatically proper couplets of a poem. For modeling the knowledge of internal tokens are defined. An example tracing the operation of model is described.*

***Key words**: artificial intelligence, affective computing, modeling, emotions, creativity, poetry.*

INTRODUCTION

Creativeness is capability to make something new and original, capability to combine and associate ideas and elements in such a way that they would form originally new connections or constructions [5,6]. Although, computers are too far away from creativity, in some occasions their behaviour looks "creative". In the cases when computers work in complex situations, especially when they cooperate with man, their creative abilities could matter significantly.

In [4] and [3] mind is looked upon as a self-organizing system, which behaviour is conceiving models for all experiences and following them. The lateral thinking has its creative potential, because it offers opportunities for exiting the constructed model, broadening the notions and opening new prospects. In [4], the method of the casual word and the method of expressing a provocative thought are described as technics for revealing new conceptions by building up accidental events. In order to generate new ideas, the associations and the functions of the word-stimulus are followed or some of its aspects – as metaphor are used in different methods for advancing to the new concept: focusing on the difference, deducing the principle, seeking after some value, describing the situation and so on.

A generalized net model of the composing verses process is presented in this article. External tokens, that have the ability of self-transforming, self-changing and self-choosing according to the key terms, received on the net-entrance, are defined. These tokens leave the net as newly created verses. Internal tokens for the existing analogies and grammatical examples are also formed. They are used for modeling the system knowledge of the existing conceptions and word combinations. The stages are: choosing the initial concepts and word combinations according to the in-coming key words, constructing grammatically correct sentences, re-arranging, selecting or rejecting some verses until a completed poem is created. Requirements for the emotional value, number of syllables, rhyming and sounding are taken into consideration. New variants are repeatedly chosen by the feedback connections. If there are no requirements for rhyming or for the step of verses, completed phrases on the given themes would be constructed.

GENERALIZED NET MODEL OF CREATING VERSES

The process of creating verses is based on the principle of assigning key words with defined emotional characteristics and level of understanding, number of syllables in a row in defined boundaries, requirement for rhyming, rhythm and melody and point of view. The process is a subsequence of next four stages: (1) deriving the proper words and word combinations according to the initial requirements (r1); (2) constructing grammatically correct and semantically proper phrases according to the grammatical patterns (r2); (3) obtaining muted grammatically correct phrases from the already available ones by substitution with words, analogical by meaning, and by suffling, shortening or expanding the phrases according to the existing grammatical patterns (r3); (4) wording the poem couplets in accordance with the requirements for point of view, emotional characteristic,

syllable number and rhymes. These stages are represented (fig. 1) by A set of net transitions.

$$A= \{r_1, r_2, r_3, r_4\}.$$

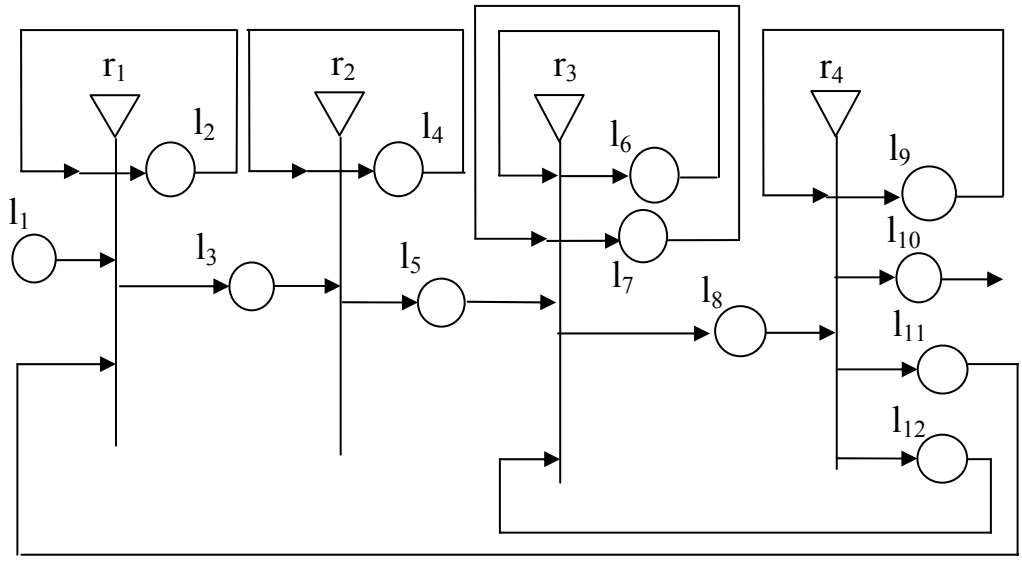

fig.1

Two types of tokens circulate in the model. The internal tokens move only in the net and model the individual knowledge of each net for concepts, characteristics and grammar rules. The external tokens are the initial words and phrases that enter the system. Passing through the net, they are transformed to phrases, some of which are selected as most appropriate for obtaining the final verses. The inappropriate tokens fall out of the net.

Regardless of the substantial difference between internal and outside tokens, each token will be presented by the triple

$< X, \Phi, b >$, where:

X is a set of initial characteristics of the token,

Φ is a function (let us call it transforming one), which defines a new characteristic h_i^* for each token i with characteristic h_i upon passing the token from the input into the output place of each transition j simultaneously with getting of the token k through this transition, with characteristic h_k.

$h_i^* = \Phi (r_j, h_i, h_k)$

b is the maximum number of characteristics, which one token can obtain during its motion in the net.

As a result of these considerations and according to [1], we receive the following generalized net:

$E = << A, \pi_A, \pi_L, c, f, \theta_1, \theta_2 >, < K, \pi_K, \theta_K >, < T, t^\circ, t^* >, < X, \Phi, b >>$

Where the rest of the designations mean:

π_A: $A \rightarrow N$, where $N=\{0, 1, 2,\} \cup \{\infty\}$ priority of the transitions.

π_L : $L \rightarrow N$, where $L=pr_1 A \cup pr_2$ and pri X denote the i-projection of the n-dimension set X, $n \in N$, $n \geq i$, $i \leq n$ (L is a set from all places of GN)

c - function, giving the capacity of the places. c: $L \rightarrow N$

f - function which defines the true value of the predicates

θ_1 - is a function assigning the subsequent moment from time, in which a given transition can be activated. The value of this function is recalculated in moments, when the active condition of the transition is over. $\theta_1(t)=t'$, where $t, t' \in [T, T + t^*]$ $t \leq t'$

θ_2 - is a function giving the duration of the active condition of the transition. $\theta_2(t)=t'$, where $t\in[T, T + t^*]$ and $t' \geq 0$. The value of the function is calculated when the transition starts functioning.

K is the set of the tokens of GN. It can be represented in the form: $K = \bigcup_{l \in Q^l} K_l$

Where K_1 is the set of the tokens of GN, which are waiting in front of place 1 and Q^l is a set of all input places for GN.

π_K is a function assigning the priorities of the tokens, i.e.π_K:K→N;

θ_K is a function assigning moments from time, in which a given token can enter, the net, i.e.

$\theta_K(\alpha)$ = t, where $\alpha \in K$, $t \in [T, T + t^*]$

T - moment from time, in which GN begins to function.

t° - elementary time step;

t* - duration of the functioning of GN.

The indexed transition matrixes are shown in fig. 2.

$r_1 =$

	l_2	l_3
l_1	$w_{1\,2}$	**false**
l_2	**true**	$w_{2\,3}$
l_{11}	$w_{11\,2}$	**false**

fig. 2 a)

$r_3 =$

	l_6	l_7	l_8
l_5	$w_{5\,6}$	$w_{5\,7}$	**false**
l_6	**true**	$w_{6\,7}$	**false**
l_7	**false**	**true**	$w_{7\,8}$
l_{12}	$w_{12\,6}$	$w_{12\,7}$	**false**

fig. 2 c)

$r_2 =$

	l_4	l_5
l_3	$w_{3\,4}$	**false**
l_4	**true**	$w_{4\,5}$

fig. 2 b)

$r_4 =$

	l_9	l_{10}	l_{11}	l_{12}
l_8	$w_{8\,9}$	**false**	**false**	**false**
l_9	**true**	$w_{9\,10}$	$w_{9\,11}$	$w_{9\,12}$

fig. 2 d)

The predicates of the transition conditions are:

IS – initial state. It is determined by the key words presented on the net-entrance.

M_initial_aggregate – method, that "catches" these circulating in transition first internal tokens, that corresponds to the external tokens – conceptions (for example emotions, semantic, pragmatic, aesthetic knowledge) and word combinations.

V_sufficiency_ia – boolean variable that defines if enough initial concepts are selected.

V_sufficiency_an– boolean variable, that defines if enough analogies are selected.

V_sufficiency_rec – boolean variable, that defines if enough reconfigurations are made.

V_sufficiency_res – boolean variable, that defines if there are enough verses for a final result.

V_pattern – boolean variable, that is "true" if the constructing of grammar expressions is allowed.

V_available_res – boolean variable that shows if there are internal tokens in the forth transition.

M_pattern – method for constructing grammatically correct expressions in accordance with the internal tokens in the second transition.

M_analogy – method for finding words, analogical by meaning and emotions the already chosen ones in the first transition and their substitution.

M_reconfig – method for rearranging, shortening or expanding the phrases in accordance with the available knowledge and grammatical patterns.

M_revaluation – method for new evaluation of the emotions after the substitution with analogies. Every word has emotional property[2] and to direct the generation towards of really desirable text we can arrange internal tokens accordingly characteristic h_i* for each token.

M_selection – method for emotional evaluation, for checking the accordance of the last syllable and the syllable number of the obtained expressions. The generation of sentences can be directed by using the token's history and table of history, which contains only "suitable" sentences.

M_humour – method for checking the familiar associations of the selected words in order to avoid black humour. This is last sentences checking for unsuitable associations.

M_sound – method for selection of the more appropriate in sounding word or phrase when there is coincidence of the emotional value, the number of syllables and the condition for rhyming with another phrases.

F_print_exit – ends the net work and visualizes the result.

The transition dependences in the model are presented below.

W1_2= IS $\wedge$ ¬V_sufficiency_ia $\wedge$ M_initial_aggregate; - initial concepts are selected and when their number is enough, the value of the variable V_sufficiency_ia is set to 'true'.

W2_3 = V_sufficiency_ia$\wedge$ V_pattern; - the two boolean variables are set to 'true'.

W11_2 = ¬V_sufficiency_ia$\wedge$ M_initial_aggregate; - if the value of V_sufficiency_ia is 'false', then the method M_initial_aggregate starts.

W3_4 = V_pattern $\wedge$ M_pattern; - grammatically correct phrases are constructed.

W4_5 = ¬V_sufficiency_an $\wedge$ ¬V_sufficiency_rec; - the values of the variables V_sufficiency_an and V_sufficiency_rec are set to 'false'.

W5_6 = ¬V_sufficiency_an$\wedge$ M_analog; - when the value of the boolean variable is set to 'true' this means enough analogies are found.

W5_7 = ¬V_sufficiency_rec $\wedge$ M_reconfig; - shuffling, shortening or expanding the parts of the sentences until the Boolean variable is set to 'true'.

W6_7 = V_sufficiency_an $\wedge$ M_reconfig; - if the value of the Boolean variable is 'true', the method for phrases reconfiguration can be called.

W7_8 = V_sufficiency_rec $\wedge$ ¬V_sufficiency_res; when enough shufflings, shortenings or expandings of expressions are made, the value of the Boolean variable for result sufficiency is set to 'false' and it can be proceeded with the fourth transition.

W8_9 = ¬V_sufficiency_res $\wedge$ M_selection $\wedge$ M_humour $\wedge$ M_sound $\wedge$ M_reconfig $\wedge$ V_available_res.

W9_10 = V_sufficiency_res $\wedge$ F_print_exit.

W9_11 = ¬V_sufficiency_res $\wedge$ ¬V_sufficiency_ia; - if the achieved result is not satisfactory, a passing to the first transition could be made, as the value of the V_sufficiency_ia variable is set to 'false'.

W9_12 = ¬V_sufficiency_res $\wedge$ V_sufficiency_ia $\wedge$ (¬V_sufficiency_an $\vee$ ¬V_sufficiency_rec);

If the final result is not satisfactory and there are enough initial tokens, then the value of one of the Boolean variables (for finding new analogies or for reconfiguration of the obtained expressions) is set to 'false'.

MECHANISMS USED IN THE PROGRAM REALIZATION

The token is modeled as a component with characteristics, methods and events. The main model methods are: for choosing the key concepts and word combinations; for constructing grammatically correct expressions in accordance with the available grammar patterns; for substitution of words and word combinations with analogies; for shortening, rearranging in accordance with another pattern or expanding an expression or word combination; for evaluation the emotional value of the obtained phrases; for avoiding broad humour; for selecting from the obtained poems according with the requirements for rhymes and syllable number in a row.

The external tokens are brought in by the user and contain information for the structure of the sentences in the poem. They contain some compulsory words or the grammatical form of the words that would construct the sentences. For example, four external tokens for constructing a quatrain could be: *Object verb moon; Adjective object I'm alone; Cloud verb gentle object; Wind verb noun distant noun.*

The words are searched for in a table for defining their root and grammatical form, the group of belonging and the group of generalization. The generalization table contains characteristics that allow a simple sentence to be written in fanciful syntax. A complicate sentence could be constructed by unspecified number of simple sentences. The thought may consist of unspecified number of sentences. The table columns can contain things that could be found in a simple sentence and pointers to the previous or next sentences. Lists of groups are presented in every cell. Each sentence is related with all the columns in a row. The lists are open for taking out or adding word groups. The group structures are also open. The characteristics of each word could be infinitely enriched. Throughout the constructing process of the sentences, the most appropriate words and expressions from the generalization table are selected, as belonging to the group list of a given group is searched for in a first place, and after that concrete words from this group are chosen.

The method get_grammar reads the external tokens and defines the group of belonging of the concretely given words and the generalization group with expressions for this word. The method Get_Sentence (arguments) is called. This method takes as many arguments as the number of columns in the generalization table (15). Some of these fields will be specified for the given sentence, and the rest, which are not necessary in the case, will be marked with dashes. The specified positions must have values in the generalization table, and the ones with dashes are ignored by the program.

The Get_Sentence_x method reads the generalization table and gets the groups for each of the specified words. The compulsory words or groups must be present in the read table row. If any of the compulsory words is missing, the table row is skipped. When the row necessary for constructing the given type of sentence is found, the real construction begins. For each meaning argument the group's list are read and the words in each group are read. Words are being read one by one. When the first word from the first group for the first argument is read, it is being saved, and the reading of word from the groups list for the next argument starts, and so on. In this way only one word for every word in the sentence is read and one sentence is constructed. The process looks like counting in different basises for each position because there could be different number of words in every position in the sentences.

Tens of sentences could be constructed by each type external token. They are kept in a history table. The syllables are counted and the last syllable is saved. After that the method of the Styx class is called. It builds up quatrains taking into consideration the order of incoming of the external tokens, the number of syllables and the number of rhymes.

The result, achieved after the bringing of the above presented initial tokens in, is:
Pass the window shines the moon
In the room I'm not alone.
Cloud gives its gentle palm.
Wind is bringing distant tone.

CONCLUSION

A generalized net model of the composing verses process is presented in this article. External tokens, that have the ability of self-transforming, self-changing and self-choosing according to the key terms, received on the net-entrance, are defined. These tokens leave the net as newly created verses. Internal tokens for the existing analogies and grammatical examples are also formed. They are used for modeling the system knowledge of the existing conceptions and word combinations. The model consists of four transitions: (1) deriving the proper words and word combinations according to the initial requirements; (2) constructing grammatically correct and semantically proper phrases according to the grammatical patterns; (3) obtaining muted grammatically correct phrases from the already available ones by substitution with words, analogical by meaning, and by suffling, shortening or expanding the phrases according to the existing grammatical patterns; (4) wording the poem couplets in accordance with the requirements for point of view, emotional characteristic, syllable number, rhymes and rhythmic. Requirements for the emonational value, number of syllables, rhyming and sounding are taken into consideration. New variants are repeatedly chosen by the feed-back connections. If there are no requirements for rhyming or for the verses step, completed phrases on the given themes would be constructed.

REFERENCES:

[1] Atanassov K. (1991). Generalized nets. World Scientific Publ. Co., Singapore, 1991

[2] Budakova D., L. Dakovski (2003), Generalized net model of a program system for gathering data with a view of forming a behavior. pp. 257-260, C8; 17th Int. Conf. SAER-2003; St.Konstantin resort, Varna, Bulgaria. 2003

[3] Даниел Голман (2000), Емоционална интелигентност. Нова система за възпитание на чувствата; Кибеа 2000

[4] Edward de Bono (1992), Teach your child how to think, McQuaig Group Inc., 1992

[5] Hofstadter D. and the Fluid Analogies Research Group (1995). Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought. Basic Books, New York, 1995

[6] Picard R. (2000), Affective computing, MIT Press Cambridge, Massachusetts London, England. 2000

[7] Winston P.H. (1992), Artificial Intelligence–3rd.ed.Addison-Wesley Publishing Company 1992

ABOUT THE AUTHORS:

as. eng. Dilyana Budakova
Department of Computer Systems, Technical University of Plovdiv,
St. Peterburg Blvd. 61, 4000 Plovdiv, Bulgaria,
bdiliana13@hotmail.com

prof. d.t.n. eng. Lyudmil Dakovski
Department of Computer Systems, Technical University of Sofia,
Kl.Ohridski Blvd.8, 1000 Sofia, Bulgaria,
seven-in@bulinfo.net