

**ON USING OF FRAME-BASED SENTENCE PATTERNS
IN INTEGRATIVE COMPUTER-AIDED
LANGUAGE LEARNING SYSTEM**

Abstract. The given work is devoted to the task of compact representation of a huge number of variants of the thought expression. The task is important for building an effective linguistic environment. The solution is based on frame knowledge representation paradigm improved to represent syntactic structures.

Key words: frame, frame-based paradigm, second language acquisition, language learning system, integrative approach.

Actuality. According to Warschauer [1] there are three key classes of CALL systems connected to three language learning approaches: behavioristic, communicative, integrative [2, 3, 4]. Behavioristic systems are focused on “drill and practice” principle and did not allow enough authentic communication to be of much value [1]. Communicative systems try to involve the computer system as stimulus, not as a tutor. The features of the communicative CALL systems are as follows: teaching grammar implicitly; encouraging students to generate original utterances rather than manipulate prefabricated language; being flexible to a variety of student responses; creating an environment in which using the target language feels natural. Integrative systems are based on two technological components - multimedia computers and the Internet [1]. The main features of such systems are as follows: authentic linguistic environment simulation, skills are easily integrated, because the media makes it natural to combine reading, writing, speaking and listening in a single activity; students studying in accordance with their individual plans; focusing on the content. The typical integrative system is Dustin program developed by the Institute for Learning Sciences at Northwestern University [5, 6].

The paper [7] represents a formal description of the scenario of the scenario-based integrative language learning system. The description

is presented in a compact form and based on mathematical background. It allows to understand the specific and potential can be achieved by the system; a set of operations should be considered; phases of evolution of such systems; make a flexible architectural solution. According to [7] the scenario can be represented as a set of steps. The key element of the step is a message which passes a meaning of the event to the user. The message consists of a number of projections of the message: text, formula, video-clip, audio, image, diagram, virtual agent actions etc. There are two types of the messages: task-oriented messages which cause interrupts of the flow requiring user reaction and context-oriented messages. Context-oriented messages form a background to make a natural involvement of the user into communication based on task-oriented messages (Fig. 1).

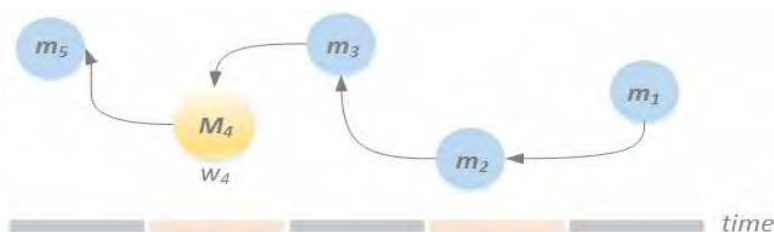


Figure 1 – The flow of the messages: context-oriented messages prepare the background for task-oriented message

Each task was defined as a 3-tuple “task definition – a set of weighted correct answers – a required time to perform the task”.

A set of weighted answers forms a pattern P^j which provides a set of pairs “expression - value”.

$$P^j =_{\text{def}} \{(s, v) \mid s \in S^j, v \in \mathbf{Q}\}, \quad P^j \subseteq \mathcal{P} \quad (1)$$

And there was also introduced an operation ψ which can estimate a certain expression against a pattern of answers.

$$\psi(s, q^j_t) = \begin{cases} v, & \exists s: (s, v) \in P^j \\ 0, & \nexists s: (s, v) \in P^j \end{cases} \quad (2)$$

Task definition. To make the interconnection between the user and the system more natural and effective means making the system flexible to a variety of student responses. This claim gets the developers faced the issue of a huge number of probable variants of acceptable answers connected to one task, considering their completeness, acceptability etc. It is obvious that this problem cannot be resolved simply by the enumeration of all the available variants. Even in connection with a

simple sentence we have a huge amount of variations: more or less acceptable, more or less complete etc. Thus, there should be applied a usable mechanism of patterns able to help teachers to describe a huge number of variants using compact structures. That is the question we try to solve in this paper.

Main part. First of all, let's define some constraints for the sentence used to patternize. In a natural dialog we can see that the complex sentences often get split into expressions of 7-10 words long using stresses and pauses. Thus, the user faces the problem of interpretation of the expression and we can assume that 7-10 words is the natural barrier of the expression to be described by the pattern.

But even considering the assumption we cannot rely on simple enumeration of the variants and face the problem of building a mechanism of compact description of the variants. Of course, this mechanism should consider both semantic and syntactic features of the expression. Using of WordNet and other similar frameworks seems not convenient.

Today, in linguistics, a method used to represent the syntactic structure of a sentence is a form of immediate constituent analysis. The method is based on the binary division principle of the structure, analysis is completed when all the constituents become indivisible. It's generic and highly useful approach in such fields as text analysis, machine translation etc., but in our case, we don't need to make such analysis, the parts of sentence can be more abstract, and this method does not give us a mechanism of compact description of the huge number of variants used to pass the meaning of the expression.

One of the powerful knowledge representation system is frame-based paradigm which first provided by Minsky in [8], and which has been refining during the last 30 years. The best analysis of the frame-based representation systems can be found in [9]. According to [10] the main goal of the frame approach was to gather all relevant knowledge about a situation in one object instead of distributing this knowledge across various axioms.

Frames. According to [9] a frame is a data structure that is typically used to represent a single object or a class of related objects or a general concept or predicate. The main benefit of such structure is its compactness. Frames reflect the human thinking mechanism and though perceived by the humans naturally. In semantic networks, properties of

the object are restricted to primitive, atomic ones. The properties in frame systems can be complex concepts described by other frames.

Slots. The slots of a frame describe attributes or properties of the thing represented by that frame and can also describe binary relations between that frame and another frame. In addition to storing values slots also contain restrictions on their allowable values.

Facets. Almost all systems specify a few other attributes for each slot besides its value such as a slot datatype and restrictions on the allowable values for the slot. Some researchers generalized this notion to allow slots to have arbitrary properties called facets of which name, value, datatype and value restriction are the usual complement.

According to frame-based paradigm, each sentence can be represented as a number of ordered slots (terminals and non-terminals) connected with the terms (variants of expression parts represented by the slots). Some slots are strongly required to be filled with the values others are optional ones. The collection of required slots forms the semantic core of the expression: the meaning cannot be passed if the slot is not filled with a proper value. Optional slots can be omitted, their task is to add some details needed to understand the message (Fig.2).



Figure 2 – Evaluation of the parts of the expression

Slots can be connected to a number of alternative values – the terms with the same meaning (i.e. synonymous). Some of the values can be regarded as preferable (the priorities can be defined quantitatively) and can be complex structures(frames) or simple – a sequence of words. The order of the slots allows us to build syntactically right sentences (formula 1). Thus, we have a semantic core of the expression represented by the frame with syntactical facets.

Frame can be defined as a set of slots (Fig.1).

$$c =_{def} \{ \} \quad (3)$$

We can describe the structure of slot as an ordered set of facets.

$$s_t =_{def} \vec{\varphi}_t, \quad \vec{\varphi}_t \in \Phi \subseteq \Phi_1 \times \Phi_2 \times \dots \times \Phi_k \quad (4)$$

Domains represent the axes of slot’s definition. For example, Φ_1 can represent the type of slot {terminal, nonterminal}, Φ_2 – number restriction used to describe the number of relationships of a particular slot that individuals can participate in etc.

We can say that two frames are similar if the sets of sentences generated by those frames are similar. Of course, here we can also talk about the coefficient of similarity.

The full structure of the frame can be represented using composite design pattern in Figure 3.



Figure 3 – Structure of the frame

Let’s look at simple example of frame.

<*Q ><*T ><!S >

Q1=as you’re aware

Q2=as you know

T1=today <*Tt>

Tt1=in the evening

S1=we’re going to be looking at your career options

The language we use to describe the frames is very closed to IC patterns improved with new semantic marks. The description is more simple and understandable than the variant of pure frame-based language description. “<...>” – defines the borders of the slot. An asterisk (“*”) placed before the denoted axis (the name of the terminal slot, e.g. Q, T, Tt, S) means that the slot is optional (“0..1” minimum number restriction), “!” before the axis declares that the slot is required (“1” minimum number restriction). The names of the slots can be selected randomly (it’s similar to declaration of the variable), their role is to represent a number of alternative values (the values of the variable).

Notably, that any part of the expression can be transformed into slot, means the selection is not connected to grammar structures (part of speech etc.). The value of the slot can be represented by the frame having some slots (Tt1 = today <*Tt>). The structure of the slot is shown in Figure 4. As we can see there are some additional facets (ts1, ts2, ts3) directed to make

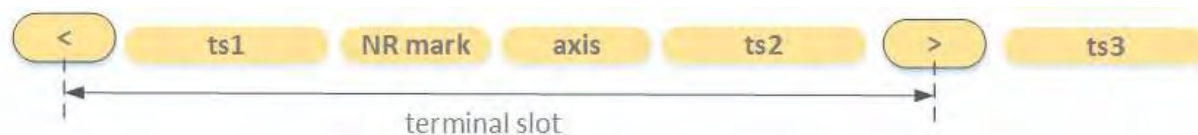


Figure 4 – The structure of the slot

The above frame produces the following set of sentences:

We're going to be looking at your career options.

Today we're going to be looking at your career options.

Today in the evening we're going to be looking at your career options.

As you know we're going to be looking at your career options.

As you're aware we're going to be looking at your career options.

As you know today we're going to be looking at your career options.

As you're aware today we're going to be looking at your career options.

As you know today in the evening we're going to be looking at your career options.

As you're aware today in the evening we're going to be looking at your career options.

It is obvious, that the adding of only one required slot with two alternative values will double the number of generated sentences.

The next question is representing a slot with “one to many” number restriction, i.e. an ability to use the values of the slot in combination. The frame below produces the following set of sentences:

<!S ><!V ><!Q >	My friend has bought an old car
S1=My friend	My friend has bought an Italian car
V1=has bought	My friend has bought an old touring car
Q1=an <!Qq&> <*Qt> car	My friend has bought an old Italian car
Qq1=old	My friend has bought an Italian touring car
Qq2=Italian	My friend has bought an old Italian touring car
Qt1=touring	car

The underlined sentences show how the mechanism works: marking the axis of the slot with “&” sign we set up additional number re-

striction facet “many” (nrMax attribute becomes assigned to 2) and the product of values made according to the order defined by the frame. This variant of interpretation is adopted to the languages where the order of adjectives is clearly defined (English, French etc.). But we have another situation when the adjectives can be used randomly. In that case we can use another sign to mark the slot showing the method of combination (nrMax in that case becomes assigned to 3).

The next question considers the rules of interpretation, e.g. according to the rule of indefinite article “a” becomes “an” before the word started with the vowel. It seems to be better to apply the rules of transformation before the loading of the frame.

How to use the patterns. The easiest way is to use the patterns for the tasks connected to translation. But, firstly, forming the material for several pairs of languages seems very expensive. Secondly, it is far from the main goal of such systems that is to simulate a linguistic environment. Another variant is to allow the user to catch the meaning of the accepted phrase and let them to pass the meaning of that phrase. It's quite often happened in real world when someone asks his/her companion to repeat the last phrase, or to explain the expression just heard on TV etc. In addition to checking the user's interpretation the system can provide an approach of evolutionary message construction. Every message has its minimal meaningful form captured by the frame. Thus, the system can show the user a process of evolution of the message from its minimal form to the complete one. It is more effective than the repetition of the whole phrase without understanding its structure and variations it encompasses.

Summary. The work provides a language to describe a huge number of sentences in a compact frame-based structure. The effectiveness of the language is based on the syntactic facets injected into the slot structure. The structure of the frame and some basic examples have also been provided. The questions of using the frames in CALL systems have also been discussed. The similar mechanism of compact representation can be used for other systems with the similar tasks.

LITERATURE

1. Warschauer M. (1996) "Computer Assisted Language Learning: an Introduction". In Fotos S. (ed.) Multimedia language teaching, Tokyo: Logos International: 3-20.
2. Wan Y. An Integrative Approach to Teaching English as a Second Language: The Hong Kong Case. Conference on English Language Learning for the 21st Century (Hong Kong, January 1996). – 14 p.
3. Lightbown P. M., Spada, N. How languages are learned. Oxford: Oxford University Press. – 1998. – 135 p.
4. Lightbown P. M., Spada, N. How languages are learned. Oxford: Oxford University Press. 1998. 135 p.
5. Warschauer M. (1996) "Computer Assisted Language Learning: an Introduction". In Fotos S. (ed.) Multimedia language teaching, Tokyo: Logos International: 3-20.
6. Wang C. On Linguistic Environment for Foreign Language Acquisition. Asian culture and history. Vol.1 No.1. 2009, pp. 58-62.
7. Litvinov A.A. On formalization of computer-aided language learning system scenario. // System technologies. – N.1(108). – Dnepropetrovsk, 2017, pp. 63-70.
8. M. Minsky 1975. 'A Framework for Representing Knowledge'. In The Psychology of Computer Vision, ed. P. H. Winston, 211 – 277. New York: McGraw-Hill.
9. Peter D. Karp, The Design Space of Frame Knowledge Representation Systems, Technical Note 520. Artificial Intelligence Center, SRI International, 1992.
10. F. Baader, I. Horrocks, and U. Sattler: Description Logics. In Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter, editors, Handbook of Knowledge Representation, pages 135–179. Elsevier, 2007.