UDC 614.2+574/578+004.38

A.A. Litvinov

ON FORMALIZATION OF INTEGRATIVE COMPUTER-AIDED LANGUAGE LEARNING SYSTEM SCENARIO

Abstract. The given work is devoted to CALL system scenario formalization aimed to generalize the experience, estimate the potential of the integrative systems, define the problems and their solutions.
Keywords: scenario, second language acquisition, language learning system, integrative approach.

Actuality. Since the first CALL (computer-assisted language learning) system appeared in 1960s, many different systems have been built, books have been written and published [1], and even academic disciplines and courses have been introduced and studied in universities worldwide [2]. Despite of a plenty of the different systems and methods existed, we can define three classes of such systems connected to three language learning approaches: behavioristic, communicative, integrative [3,4]. Warschauer [5] connected them to three phases of learning systems improvement and development: behavioristic, communicative, integrative. These systems cannot be regarded as competitors, but rather can be seen as three coexisted facets of a brilliant which effectively complement each other.

Behavioristic systems are focused on using the computer as a tutor governing by the “drill and practice” principle. A typical behavioristic system is PLATO project. It is based on vocabulary drills, grammatical rules and translation tests. However, behavioristic systems did not allow enough authentic communication to be of much value [5]. In fact, the linguistic environment for language acquisition is very important. It presses the students to communicate with other people in real time mode, brings a lot of examples of how to say something, a lot of variations of how to express a thought, gives many opportunities to experiment with the language [6]. It is very different from the drill and practice approach used within the classroom. *Communicative CALL systems* involves the computer 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; avoiding to tell students they are wrong and being flexible to a variety of student responses; creating an environment in which using the target language feels natural. Communicative CALL systems software include a number of programs which may not have been specifically designed for language learners. Computer system takes place of a passive mediator directed to connect the students and teachers, helping them to build a community.

Integrative systems are based on two technological components - multimedia computers and the Internet [5]. Multimedia resources can be used by the students on demand allowing to build an authentic linguistic environment able to affect the user through several channels of perception simultaneously (listening is combined with seeing, like in the real world). 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, honing in on particular aspects and skipping the others; focusing on the content, without focus on language form or learning strategies. The typical integrative system is Dustin program developed by the Institute for Learning Sciences at Northwestern University [7]. The program was developed for use by foreign employees of Andersen Consulting. It helps employees to enhance their English skills within the context of actual environment they will encounter when they attend business meetings at the training center. Dustin includes a number of video scenarios that all employees will encounter (airport, checking into a hotel, ordering food). User interacts with different persons from video-clips, posting his/her answers. If the answer is correct and trainee is successful with a scenario, he/she goes to the next one. But when the trainee is not successful, he/she gets notified by the system what they have to learn in a very direct way. Failure and its processing is a key element in Dustin, failing at a task make students interested in finding out what they need to know to succeed [7]. The correction of the user answers are made using a tutor agent, which in communicative user-centric manner suggests a correct way to convey the message [8]. Thus, the student feels engaged in the world of real life communications, having a clear goal, learning through apprenticeship. Another project based on the same principles is SPELL developed and discussed in the works of Hazel Morton [9,10]. The main features of the project are as follows: automatic speech recognition; embodied 3D virtual agents and virtual worlds for the creation of scenarios; focused on developing oral skills in the target language.

Task definition. Scenario is the most important component of an integrative system. The quality of the scenario and the ability of the system to play it can be seen as two key components affect the quality of the realistic language environment simulation aimed to help user to acquire language skills effectively. Secondly, the effect of the system depends on the variety and complexity of the situations represented by the scenarios. The system should be opened to add new scenarios and the scenarios should cover all the domains of knowledge (content) the users are interested in. One of the conditions to make it real, the operation of new scenario creation should be rather simple. As a result, the system will have a huge amount of scenarios and a search mechanism able to retrieve a number of similar scenarios with the properties the user is interested in, compare different scenarios etc. The third important aspect should be considered is the ability of the system to be open for multiple foreign languages learning.

To solve the given problems effectively, we should clearly understand what the scenario is, or more precisely, what the scenario may be. The way of such understanding should be based on the strong foundation of mathematical formalization. The formalization is directed to help us not only in terminological questions, but mostly in questions of systems' potential, configurations and improvements. Despite of an exceptional role of the scenario concept for integrative CALL systems, author has not found any works devoted to its formalization. Thus, the given work is intended to fill this gap.

Main part. Generally, the system of scenarios can be defined as an algebraic system composed of a set of scenarios Σ , relations Ω_r and operations Ω_o .

$$\mathfrak{S} = \langle \Sigma, \Omega_r, \Omega_o \rangle, \qquad (1)$$

Each scenario $\sigma_i \in \Sigma$ composed of two required elements: *scenario descriptor* and a none-empty, ordered or partially ordered, finite *set of steps. The set of steps defines* an order and a specific of events each of which aimed to pass a *message* to the end-user. The number of steps is limited.

Scenario descriptor is used for scenario classification. It's an element of n-ary relation on n domains-classifiers (classification axes)

$$D^{+} \subseteq K_1 \times K_2 \times K_3 \times \cdots \times K_n, \quad (2)$$

where K_i – *i*-th classifier. The typical classifiers are as follows: interest, language, skill level, rating. Classifiers can be represented as taxonomies.

In general, D^{+} consists of required classification axes denoted by D and optional ones denoted by D^{-}.

$$D = D^{+} \setminus D^{-}. \quad (3)$$

Scenarios are similar iff their descriptors' projections on required domains are equal.

$$\sigma_i \sim \sigma_j \Leftrightarrow \pi_D(\vec{d}_i) = \pi_D(\vec{d}_j), \qquad \vec{d}_i, \vec{d}_j \in D^{+}. \quad (4)$$

More precisely we can talk about a degree of similarity. Similarity operation is widely used for scenario retrieval manually by the user and automatically by the information system unit responsible for making an individual language acquisition plan.

The step of scenario reflects a real world event (more precisely an event of linguistic environment) appeared in a definite time slice. Consequently, a sequence of steps reflected a set of such events is very similar to a sequence of frames composed a movie. The key difference from the movie is the involvement of a user in the flow of events and the dependence of the flow on the user's activities. It looks like an interrupt caused by the movie and processed by the user. And in accordance with the results of the processing system chooses the next sequence of steps to play, including a switch to the alternative flow or even playing another movie.

Message. The key element of the step is the *message* responsible to pass the meaning of the event to the user. Some messages require user's reaction. A set of messages can be described by the relation on a set of domains each of which represents objects of a type or a channel. We can say that a message consists of a number of objects, or a number of projections of the message on the domains

$$M \subseteq T_1 \times T_2 \times T_3 \times \cdots \times T_m \quad (5)$$

M – the whole set of the messages registered in the system; M_i – a set of messages connected to the *i*-th scenario σ_i.

 ISSN 1562-9945

Typical examples of the message projections are as follows: text, formula, video-clip, audio, image, diagram, virtual agent actions etc. More general channels, such as audio, can be divided into sub-types such as speech audio and environment supplementary sounds (airport, cafe).

A process of passing the message to end-user is shown in figure 1. *A*, *T*, *I*, *B*, *C* – domains, and a_i, b_m, c_n, t_j, i_l – denote the projections of the message $\vec{m}_z \in M_i$ on the domains, i.e. $[\![\pi_A(\vec{m}]\!]_z) = a_i$.

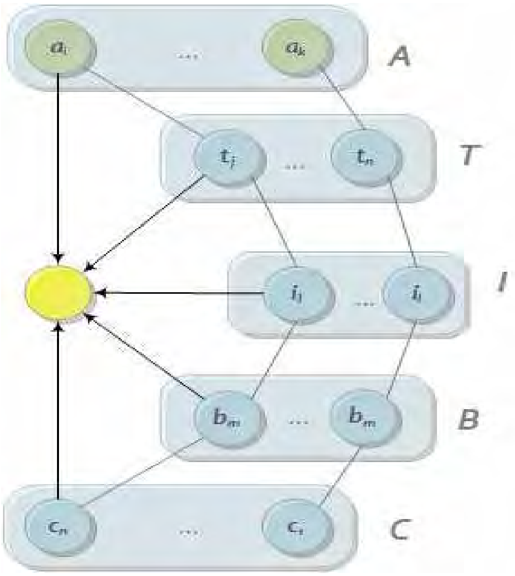

Figure 1 – A process of passing the message to end-user

Each projection of the message takes some time. Consequently, the time needed to pass the message is defined by the slowest projection.

$$\vec{m}_z =_{def} \langle (a^1_i, \ldots, a^n_j), \bar{t}_a \rangle, \quad a^1_i \in T_1, a^n_j \in T_n \ , \qquad (6)$$

$$\bar{t}_a = \max_{i=1,n}(\{\tau(\pi_i(\vec{m}_z))\}) \ , \qquad (7)$$

$\tau(\pi_i(\vec{m}_z))$ – denotes an operation of calculating a required time to pass the *i*-th projection of *z*-th message.

After the message has passed, user needs some time to perceive the message. To make the simulation more closed to the real life the time should be defined for each message individually.

There are two types of the messages: task-oriented messages which cause interrupts of the flow and require user reaction; infrastructure-oriented messages. Infrastructure or context-oriented messages form a background to make a natural involvement of the user into communication based on task-oriented messages connected to the tasks.

Tasks and reaction. A task is the most important component which provides a feed-back, allowing the system to understand the properties and specific of the user and, as a result, to make the learning process more flexible and effective. It is very important that the task-oriented message could be connected to a number of tasks each of which requires user's involvement.

Can we say that only the group of tasks composes the core of the scenario and the minimum set of required steps to pass the scenario is the set of tasks? Not obviously, because some tasks can be connected to the infrastructure messages and no one can give a correct answer without them. Consequently, the core of the scenario (minimalistic version) needed to pass the scenario by the user includes all tasks messages with the required context. Getting the minimalistic version is very important for the test-oriented users who don't want to waste their time on a soft involvement. Thus, we have defined an operation of getting a minimalistic version of the scenario.

To understand the connection between the messages and the tasks we have to separate the domains of tasks and the domains of messages.

$$\vec{a}_z =_{def} (a^1{}_i, \ldots, a^n{}_j), a^1{}_i \in T_1, a^n{}_j \in T_n \quad, \tag{8}$$

$$\vec{q}_x =_{def} (q^1{}_i, \ldots, q^m{}_k), q^1{}_i \in Q_1, q^m{}_k \in Q_m, \tag{9}$$

$\vec{a}_z$ – a message, $\vec{q}_x$ – a vector of tasks each component requires user's involvement.

Each task is a 3-tuple "task definition – a set of weighted correct answers – a required time to perform the task".

$$q^j{}_i =_{def} (\hat{q}^j, P^j, t). \tag{10}$$

A set of weighted answers forms a pattern P^j . Generally, the pattern provides a set of pairs "expression - value". An expression is not always textual and totally depends on the domain it belongs to, i.e. if the task connected with objects manipulation then the expression would be an ordered set of actions. A textual expression should be considered as a variant

$$P^j =_{def} \{(s, v) | s \in S^j, v \in \mathbf{Q}\}, \qquad P^j \subseteq \mathcal{P} \ . \tag{11}$$

We should introduce an operation ψ which can estimate a certain expression against a pattern of answers.

$$\psi(s, q^j{}_i) = \begin{cases} v, \exists s: (s, v) \in P^j \\ 0, \nexists s: (s, v) \in P^j \end{cases}. \tag{12}$$

 ISSN 1562-9945

A summary time needed to pass the tasks can be calculated in accordance with the formula

$$t_q = \sum_{i=1}^{m} \tau(q^i) \quad (13)$$

We should also define an operation $\nu(\vec{q}_z)$ which can get a vector of estimations of the tasks $q^j{}_t$ performed by the user.

$$\vec{v} = \nu(\vec{q}_x). \quad (14)$$

Thus, a message can be defined as a tuple

$$\vec{m}_z =_{def} \langle \vec{a}_z, \vec{t}_a \rangle. \quad (15)$$

And a task collection connected with the message can be defined as

$$\vec{r}_x =_{def} \langle \vec{q}_x, t_q \rangle. \quad (16)$$

A set of tuples $\langle \vec{m}_z, \vec{r}_x \rangle \in M_i \times R_i$ forms the foundation of the scenario, where R_i is a subset of the general set of scenarios $\boldsymbol{R}$ defined in the system. Index i indicates, that the subset is selected for i-th scenario.

$$M_i =_{def} \{\vec{m}_z | z \in \mathbf{N}\}, \quad (17)$$

$$R_i =_{def} \{\vec{r}_x | \forall \vec{r}_x : \exists \langle \vec{m}_k, \vec{r}_z \rangle, \vec{m}_k \in M_i, x, k \in \mathbf{N}\}. \quad (18)$$

Modes. An important operation should be considered is the function of modes switching. The function allows to transform the basic version of the scenario into a form convenient for a user, it can be regarded as a key component of adaptation mechanism made system more flexible and highly adoptable for different kinds of users. For example, it can affect the speed of events flow, including/excluding the steps etc.

$$\mu_j : \sigma_i \to \sigma^j{}_i, \quad (19)$$

μ_j - defines a transformation of j-th type which can transform the scenario σ_i into a form of $\sigma^j{}_i$. Generally, we can talk about the vector of transformations applied to a scenario in order to adopt it for the user.

Next step definition. The definition of next step performed by the γ component included in scenario playing procedure. It depends on the results of user activities estimation (see (14)). Simple variant of the procedure takes the next step from the ordered set of messages. More sophisticated one goes in accordance with the scheme included a set of alternative flows directed by the conditions based on the vector of estimations

$$\gamma : C_i \times F_i \to C_i, \quad (20)$$

where F_i – a scheme includes a set of alternative flows.

The final definition of scenario σ_i can be represented by the formulas

$$\sigma_i =_{def} \langle \vec{d}_i, m_0, C_i, F_i \rangle, \quad m_0 \in M_i, \vec{d}_i \in D, \tag{21}$$

$$C_i \subseteq M_i \times R_i, \tag{22}$$

where m_0 denotes the initial message.

Summary. The given work 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. The similar mechanism can be used for other scenario-oriented systems.

LITERATURE

1. Hubbard, P. (Ed.). Computer Assisted Language Learning: Critical Concepts in Linguistics, Volumes I-IV. London & New York: Routledge. – 2009.
2. Stanford University. Linguistic department. An Invitation to CALL. Foundations of Computer-Assisted Language Learning. [Electronic resource]. – Access mode :
http://web.stanford.edu/~efs/callcourse2/CALL1.htm
3. 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.
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. – P. 58-62.
7. Schank R., Cleary C. Engines for education. Volume 2. Routledge.–1995.–248p.
8. Schank R. Inside Multi-Media Case Based Instruction. Routledge.–1998.–472p.
9. Morton H., Jack M. Scenario-Based Spoken Interaction with Virtual Agents. Computer Assisted Language Learning.–July 2005.–Vol.18,No.3.–P. 171 – 191.
10. Morton H., Gunson N., Jack M. Interactive Language Learning through Speech-Enabled Virtual Scenarios. Advances in Human-Computer Interaction. 2012 [Electronic resource]. – Access mode :
https://www.hindawi.com/journals/ahci/2012/389523/

ISSN 1562-9945