Algorithmic Interactive Presentation of Notions

Pavel S. Pankov, Sagyn A. Alaeva, Vasilii A. Kutsenko

International University of Kyrgyzstan, Kyrgyz National University,
Kyrgyz-Russian Slavic University, Bishkek, Kyrgyzstan

Abstract

A kind of meta-language is proposed to present notions of natural languages in interactive form and to learn elements of languages on computer without any other language being a medium by means of the user’s arbitrary actions by computer mouse with feed-back. Such actions by means of the «Drag-and-Drop» mechanism and the notion of «Active point» are implemented.

1. Introduction

The authors [1] demonstrated similar things having various attributes together with calling them in any artificial «language» (with nouns and adjectives only) to children. If the child called other things in this «language» properly then s/he was asked why s/he had used these words. Winograd [2] proposed giving commands to a robot with such words as «table», «box», «block», «pyramid», «ball», «grasp», «moveto», «ungrasp». By using these ideas and opportunities of up-to-date computer equipment, we [3] propose software for learning any language without any other language being a medium, with the following principle: guessing and fulfilling the only natural action possible in the environment.

That is, the pupil begins with thinking in the learned language. (The only use of the pupil’s native language in proposed software is a small instruction in using the computer equipment for input (mouse, joystick, trackball, gloves, microphone …) and output (display, loudspeaker, earphones, stereoscopic glasses …) to treat (take, move, put …) virtual objects (the simplest media are familiar already for most users nowadays).

To substantiate this approach we introduced [4] and implemented [5] the following

Definition 1. Let any «notion» (word of a language) be given. If an algorithm acting at a computer performs (generating randomly) sufficiently large amount of situations covering all essential aspects of the «notion» to the user; gives a command involving this «notion» in each situation; perceives the user’s actions and performs their results clearly on a display; detects whether a result fits the command, then such algorithm is said to be a computer interactive presentation of the «notion».

(Certainly, commands are to contain other words too. But these words must not give any definitions or explanations of the «notion»).

The first version of the software won a bronze medal at Infomatrix – The International Computer Project Competition in Bucharest, April 2004.

An implementation of this definition is proposed in this paper.

Demonstration of the improved version of the software at the conference is supposed.

2. Mathematical environment

Let D be a rectangle (display), each point (pixel) of it has a color. One of colors is one of background, others are ones of objects. A complex object is a family of connected subsets of D such that they are constant or varying continuously while time.

An object has a position (coordinates of its left upper corner) and can also have other «attributes»: color, size, direction (angle of rotation), image, caption, sounding etc.

An object can also be changing itself and be changed (shifted, rotated, pushed …) by the user.

Objects can overlap each other. A complex object is a set of some (two-three) simple objects which overlap each other and form a connected set.

Remark. From this standpoint, while using the mechanism «Point-and-Click» each object is represented with the only set. The mechanism «Drag-and-Drop» involves parallel shifts. The notion of «active point» [6] gives the opportunity of rotation and other transformations.

Definition 2. An «active point» is a little spot (preferably of red color) within an object. Moving it by means of computer mouse moves the object itself with any transformation which sense can be guessed by the user after some motions.

3. Algorithmic language for presentation of «notions»

The proposed algorithmic language contains:

- a formalized subset of a natural language;
- statements describing environment: objects and relations among them;
- statements describing the user’s opportunities;
- statements describing conditions to meet a statement in the natural language.

Remark. We do not consider sounding of commands written in a formalized subset of a natural language because it is well-known.

The «notion» consists of:

- command written in the subset of the natural language, with randomizing;
- description of environment;
- description of the user’s opportunities;
- description of conditions to meet this command.

Remark. We do not give a formal description of such Algorithmic language because the score of statements depends on actually developed software and the correspondence between the command in a natural language and consecutive descriptions is organized by the programmer due to the sense of «notion».
4. Consequence of «notions»

By this approach, the set of «notions» is semi-ordered: of each two «notions» either one must be performed before other or they may be introduced independently. Particularly, transitive verbs can be subdivided into «independents» of essence of direct objects (for instance, move, take, put, find, hide) and «dependents» (for instance, write, paint, flex, read).

Correspondingly, «independent» verbs can be learned with any nouns but «dependent» verbs are to be learned after some fitting nouns and/or adjectives.

5. Content of the software

Described software consists of two basic parts: “Tasks constructor” and “Player for learning and testing”. The first part is supposed to be used as a field of individual learning programs. While using “Tasks constructor” it is possible to create and correct objects, actions over the objects, as well as tasks (which depends on the age of trainee, purposes of the training etc.). These options help to extend the list of the software’s possibilities.

The second part is created to implement training process itself. So the first part is intended for the limited range of users-programmers (the user should have special knowledge to be able to organize training process), the second part can be used by trainees directly.

The proposed software has two modes: learning and testing. At the first stage the user fulfills commands given by a loudspeaker (earphones) and duplicated in written form by a display. At the second stage the user gives commands by voice and the program tries to recognize and to fulfill them.

The learning mode proposes following tasks successfully: Initial (I-)task demands the only natural action in the initial Environment;

Consecutive (C-)tasks demand alternative actions in supplemented Environments;

Repetition (R-)tasks demand actions in preceding Environments.

Nouns are introduced in I-tasks as the only objects which corresponding words can be referred to; they are introduced in C-tasks as alternative appearing objects.

Verbs are introduced as the only natural actions in the Environments treating the mentioned objects.

Adjectives are introduced in I-tasks as the same properties of two different objects; they are introduced in C-tasks as properties of alternative appearing objects.

Prepositions, cases and postpositions (in Turkic languages) are introduced implicitly while new Environments appear.

Also, all tasks are generated randomly: if the software is run again then, in the learning mode, tasks at the same level would be others but of the same patterns (that is, in the testing mode, they would be of the same difficulty).

According to Windows software customs, in the learning mode:
- if the user tries to begin a wrong action then the computer permits a small shift only and restores the starting-position;
- if the user tries to finish an action wrongly then the computer also restores the starting-position (number of attempts is not bounded);
- at each moment the user can pass to the next task or to a preceding task or to the main menu.

Successful actions are encouraged by pleasant music and sayings.

The testing mode offers R-tasks only. When the user’s action is complete the software announces the mark: «good» or «bad» and the next task is offered.

When all tasks of the chosen level are taken the total mark is announced.

6. Examples

Remark. All the examples given below can be written in the proposed Algorithmic language but we suppose that concrete examples (with keeping random permutations in mind) are more understandable.

6.1. «Independent» transitive verbs

We describe consecutive introducing of transitive verbs (nouns occurring are auxiliary).

Below Ball, Cube and Pen overlap Box, Table and Chair. Corresponding cases in Kyrgyz and other Turkic languages are also mentioned.

1) «Put»: Environment: Ball; Box. I-task: Put the ball [Accusative case] into the box [Dative case]. (The trainee is to fulfill the only action because there are no other possibilities).

2) Environment: Ball; Box, Chair. C-task: Put the ball onto the chair (The quick-witted trainee is to fulfill this action because the order is changed and this is the only alternative action).

3) Environment: Ball, Pen; Box. C-task: Put the pen onto the chair.

4) «Make». Environment: Little part of Triangle; Main part of Triangle. I-task: Make the triangle [Accusative case].

5) Environment: Little part (both of Square and of Triangle); Main part of Triangle, Main part of Square. C-task: Make the square.

6) «Take». Environment: Ball on Table; Box. I-task: Take the ball [Accusative case] from the table [Ablative case] and put it into the box [Dative case].

7) «Left». Environment: Pen on Left Table; Left Box, Right Table, Right Box. I-task: Take the pen from the left table and put it into the left box. (The quick-witted trainee is to fulfill this action because the word «left» has been sounded twice).

8) «Blue». Environment: Blue Ball; Blue Box; White Box. I-task: Put the blue ball into the red box. (The quick-witted pupil is to fulfill this action because the word «blue» has been sounded twice).

9) «Move». Environment: Square on Box; Ball (not seen at first) in Box; Table. (Square overlaps Box and Ball). I-task: Move the square [Accusative case] to the left; take the ball from the box and put it onto the table.

10) «Open». Environment: Triangle on Box; Pen (not seen at first) in Box; Chair. (Pen overlaps Box and Chair; Triangle overlaps Box and Pen). I-task: Moving the triangle open the box [Accusative case]; take the pen from the box and put it onto the chair.

11) «Find». Environment: (About ten) Boxes; Squares on Boxes; Pens (not seen at first) in some Boxes; Ball (not seen at first) in one of Boxes; Table. (Squares overlap Boxes, Pens and Ball). I-task: Opening the boxes find the ball [Accusative case] and put it onto the table.

12) «Turn». Environment: Stick (vertically) with (red) Active Point in its upper end; Three segments forming a square without lower edge. I-task: Turn the stick [Accusative case] left by the red point and make a square.
6.2. «Dependent» transitive verbs

13) «Ring». Environment: some Bells of different colors; Ball. I-task: Ring the green bell [Accusative case] with the ball [Nominative case with postposition].


15) Environment: Red Brush, Yellow Brush; two Green Cubes randomly, White Cube, Green Triangle, Red Ball. R-task: Paint a green cube with a yellow brush. Paint the red ball with the yellow brush. Paint the yellow cube with the red brush. Paint the green cube with the red brush.

6.3. Articles (only for languages with articles: English, French, German …)

16) Definite article «The». Environment: Some Cubes in a file and a single Cube; Box. I-task: Put the cube into the box.

17) Indefinite article «A». Environment: Some Cubes randomly; Box. C-task: Put a cube into the box.

18) Environment: Ball, Cube, Pen; Table. C-(complex)-task: Put a thing onto the table.

19) Environment: Two Cubes randomly; Box; Table. C-(complex)-task: Put a cube into the box. Put this cube onto the table. Put that cube into the box.

6.4. Example of complex notion

Environment and general Purpose (both by voice and written) are performed. The trainee is to attain Purposes by means of a sequence of actions of learned patterns. Such sequence can be not unique, and the final result only is scored.

Example of a transitive verb:

20) «Gather». Environment: Ball on (very little) Left Table, Book in Box, Book on Blue Chair, Yellow Chair, Right Table. I-(complex) task: «Gather two books on the left table».

One of sequences of actions:
Take the ball from the left table and put it onto the right table. Take the book from the box and put it onto the left table. Take the book from the blue chair and put it onto the left table.

6.5. Tenses

21) Past Indefinite Tense, Future Indefinite Tense. Environment: Turtle in Open Box; some Open Boxes; some Closed Boxes. Turtle goes out of Open Box and moves slowly to one of Closed Boxes. I-(complex) task: «Close the box where the turtle was; open the box where the turtle will be».

Also, some parents used elements of this methodic (without computer) for children of 3-4 years old.

8. Conclusions

We hope that such software would be able to be used as an introduction to learning languages by means of CD-ROMs. Also, the test mode can be used in various kinds of examinations and competitions for all students learning foreign languages.

We also hope that developing the Algorithmic language and minimizing descriptions of «notions» would promote investigation of primary meanings of words in various languages.

9. Acknowledgements

We would like to thank the Adam Mickiewicz University for the support of the first stage of this investigation.

10. References


