Development of a computer system to support knowledge acquisition of basic logical forms using fairy tale "Alice in Wonderland"

Antonija Mihaljević Španjić*, Alen Jakupović*, Matea Tomić*
*Polytechnic of Rijeka, Business Department, Rijeka, Croatia
amspanjic@veleri.hr
alen.jakupovic@veleri.hr
mtomic@veleri.hr

Abstract - The knowledge acquisition of basic logical forms (term, statement, conclusion, and proof) is a prerequisite of the acquisition of any other knowledge and skills. Therefore, the knowledge acquisition of basic logical forms needs to be addressed in the earlier stages of education. In order to support the acquisition of knowledge and skills in the domain of logical forms, the authors of this paper propose the development of a computer based system for the presentation of thematic information, tasks and games based on the parts of "Alice in Wonderland" – fairy tale recognizable and popular due to Carroll's game with the basic forms of logical thoughts. The computer system would allow children and adults to gain the subject knowledge. A paper describes two initial phases of the aforementioned system - analysis and design. Analysis specifies in detail the basic system requirements, expressed in terms of learning outcomes. Design specifies in detail thematic information, tasks and games which will try to ensure achievement of learning outcomes specified in the analysis phase. The focus of design is in the use of a fairy tale Alice in Wonderland, and in the development of a computer based system.

I. INTRODUCTION

Logical forms indicate forms of human thought. The basic logical forms of thought are the term, statement, conclusion and proof. The term is encountered when there is a confrontation with an unknown object. In such a situation arises the need of understanding the meaning of that object and creating a clue about it. During a life and learning a person constantly acquires new concepts and expands his knowledge. In addition to directly perception of the object and acquisition of terms, it is possible to indirectly perform some conclusion on the basis of what is known about the object or event. When we want to confirm or refute a statement, this is made by use of proof, i.e. by application of rules of consistent logical reasoning. From the foregoing is evident that logical forms are in constant daily use and how much they are significant. Logical forms are subject to interest of logic as a scientific discipline that investigates shape or form and the ideal rules of thought. Logic and its subject logical forms are prerequisite for any human scientific or professional activity [6], [10], [11].

The intention of the author of this paper is to develop a computer system that would support the acquisition of knowledge and skills in the domain of logical forms. It would be available to a wide range of students, and would be tailored to the different age groups of students. The basic instructional method implemented in a computer system would be problem-based learning [1], [2], [5], [7]. The traditional method of teaching is subject-based learning. Figure 1 shows the basic difference between the two mentioned learning approaches [5].

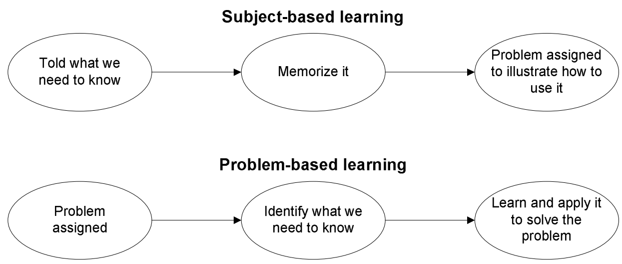

Figure 1. Difference between subject and problem based learning

In subject-based learning teacher provides content that is needed to learn. Students acquire the given content. Following the adoption, students are given problems to which they should apply the acquired knowledge.

In problem-based learning teacher first pose a problem to students. Guided by the problem, the students, together with a teacher, determine the content that is needed to resolve the problem. Then students adopt and apply identified content to a given problem. Problem-based learning is active, integrated, cumulative and connected [1].

Development of a computer system to support the acquisition of logical forms is based on the parts of "Alice in Wonderland" - fairy tale recognizable and popular due to Carroll's game with the basic forms of logical thoughts [4]. Also development of the system follows the ADDIE model, which consists of five phases: Analysis, Design, Development, Implementation and Evaluation [8], [9].

In addition to introduction, the paper consists of three chapters. In the second chapter - Analysis phase of system development to support the acquisition of logical forms - are shown the learning outcomes that students should achieve, and for which the system will provide support. The third chapter - Design phase of system development to support the acquisition of logical forms – shows examples of the teaching content and the way they are presented to students by taking into account the problem-based learning method. The paper ends with a conclusion about the directions for further development of computer based system to support the acquisition of logical forms.

II. ANALYSIS PHASE OF SYSTEM DEVELOPMENT TO SUPPORT THE ACQUISITION OF LOGICAL FORMS

According to the ADDIE model of system development, analysis phase should determine the components that are needed in other phases of the system development. Basically we can say that the analysis should answer the question of what developed system must do, or which requirements should be satisfied by its work. Especially for the development of a system which should support the learning, analysis should provide answers to the following questions [8], [9]: Who is the learner? What is the instructional goal of the lesson? What are the delivery options? What instructional technology, if any, would enhance student learning? What will the students do to determine competency? What is the timeline? What are the online pedagogical considerations?

Computer based system to support the acquisition of logical forms should be used by learners of all ages, starting from elementary school age. In the first phase of system development focus will be on the students, because the authors of the paper participate in faculty teaching.

The learning outcomes whose achievement we want to be supported by the system are shown in Table 1. Stated are the learning outcomes for each logical form (term, statement, conclusion, and proof). In their definition has been applied Bloom's taxonomy of learning domains [3].

Computer based system to support the acquisition of logical forms should be available to the wide range of

TABLE I. THE LEARNING OUTCOMS FOR LOGICAL FORMS: TERM, STATEMENT, CONCLUSION AND PROOF

		LOGICAL FORMS			
		Term	Statement	Conclusion	Proof
BLOOM'S TAXONOMY OF LEARNING DOMAING	Knowledge	To recognize a word or sequence of words that signify the term in a given text or group of objects. To define the term.	To recognize simple statement and logical operators in a complex statement. Mark them by symbols (if they are written in some natural language).	To recognize simple logical elements in given logical form and classify them in the appropriate group – statement symbols and logical operators.	To recognize the logical forms of statements and conclusions in a given text and classify them in the appropriate group – the conclusion premises.
	Comprehension	To extract / identify higher and lower terms from the pair of terms or from a given text.	To appoint type of given statement and logical operators.	To identify the type of logical form - hypothetical syllogism, modus ponendo ponens, modus tolendo tollens, disjunctive syllogism, modus ponendo tollens, modus tolendo ponens.	To identify basic rules of conclusion in a given text using which one can prove a given statement starting from the premises.
	Application	To show given group of terms according to the range size.	To show simple statements and logical operators from which can be built complex logical statements.	To find logical statements and logical operators in a given text.	To mark by symbols all the logical forms – statements, conclusions and rules of conclusion by which it can be proved given statement starting from the premises.
	Analysis	To extract all terms from the given group of terms or text which are in certain logical relationship.	To extract all the simple logical statements and logical operators from the given set and by which one can build a certain type of complex statement. To extract all statements from the group of the statements or default text that are in some relation to logical square.	To extract all logical statements and logical operators from a set of logical statements, using which one can build a given form of conclusion.	To group logical forms of statements and conclusions, and the rules of equivalence that will be used in a proof.
	Synthesis	To find terms that are in corresponding relations with the given term and to build a table or devise a similar construction.	From the set of simple logical statements and logical operators to build a logical form of conclusion in a natural language. From the set of statements to select those statements that are in some relation to logical square.	To compile a logical form of conclusion by sequencing logical statements and link them by logical operators.	To build a proof using grouped logical forms, and basic rules for proving a given statement starting from premises.
	Evaluation	To reassess / show the relationship between given terms in Venn diagrams.	To check the validity of statement using a method of *reductio ad absurdum* and a boolean table.	To check the validity of conclusion using a method of *reductio ad absurdum* and a boolean table.	To build a new proof for the same statement based on the same premises, applying different rules and compare the existing proof with a newly built.

learners and that in the online and offline work mode. Availability should increase in the way that system can be used by desktop and laptop computers, but also by tablet computers and smart phones. It is assumed the minimum computer literacy of learners.

The system should support the learner-content interaction, i.e. without the instructor. This interaction will be supported with the instructional method of problem-based learning. To learners should be set a problem which they are trying to solve. In case they do not have enough knowledge for the solution of the problem, learners should be given the content that will allow them to adopt the necessary knowledge.

Through the established system learners should have the possibility to check the acquired knowledge. This should be realized in a way that they are given the problem which they are trying to solve – similar to the way in which the problem is set before acquiring knowledge. The difference is that now they do not have available content with the knowledge necessary to solve the problem.

The interaction of system and learners should be accomplished in several ways (for example by entering data, marking, positioning, choice, drag and drop, etc.).

III. Design phase of system development to support the acquisition of logical forms

Results of the analysis (the first phase of ADDIE model) enter the second phase of system development – the design phase. This phase needs to answer the question how the developed system would satisfy the requirements which are identified in the analysis phase. Especially for the learning supporting systems, design should provide answers to questions [8], [9]: which assignments student needs to perform in order to attain certain learning outcome specified in the analysis phase, which is targeted performance objectives of the student (what the student will do, under which conditions, and which the result is acceptable) and the manner in which will be tested the achievement of certain learning outcomes.

Not diminishing importance of responses to other questions, the authors of paper in this section provide an answer to the first question - description of the tasks that student needs to perform in order to reach the learning outcomes. This is to emphasize that system design wants to support problem-based learning. The logical form conclusion and its learning outcomes that fall in all learning domains (see Table 1) will be used as example (it includes all other logical forms). Descriptions have the following structure:

1. learning domain – learning outcome
2. description of the task which student needs to perform in order to achieve these learning outcome and its correct solution - the problem description
3. content that will be available to the student to solve the assigned task (problem) and reach a defined learning outcome.

In tasks for learning domain application, analysis, synthesis and evaluation, concrete examples of fairy tale "Alice in Wonderland" are used. In order to clearly present the learning outcomes, the corresponding description of the task, its solution and related content, the table view will be used. In analogy it can be created description also for other logical forms - term, statement and proof.

TABLE II. Design for logical form "Conclusion" – learning domain "Knowledge"

Learning domain: learning outcome
Knowledge: To recognize simple logical elements in given logical form and classify them in the appropriate group – statement symbols and logical operators.
Task – solution
From displayed logical form should be recognized logical elements and classify them in the appropriate group. $(X \land Y) \rightarrow \neg Z$ Statement symbols: $\neg\neg Z$ Logical operators: $\neg(X \land Y)$ **Solution:** Statement symbols: X, Y, Z Logical operators: ∧, →, ¬
Content
Logical forms consist from statement symbols, and may contain logical operators. Statement symbols are used in the labelling of simple statements, and are used capital letters (e.g. A, B. .., X, Y, Z...). Logical operators are used for composing simple statements into more complex statements or for their negating. Logical operators are: conjunction ∧, disjunction ∨, implication →, equivalence ↔, and negation ¬.

TABLE III. Design for logical form "Conclusion" – learning domain "Comprehension"

Learning domain: learning outcome
Comprehension: To identify the type of logical form - hypothetical syllogism, modus ponendo ponens, modus tolendo tollens, disjunctive syllogism, modus ponendo tollens, modus tolendo ponens.
Task – solution
Which type is the given logical form. $(X \land Y) \rightarrow \neg Z$ $\neg\neg Z$ $\neg(X \land Y)$ **Solution:** Given logical form is type of modus tolendo tollens.
Content
There are six basic forms of conclusion that can be recognized according to the forms of premise and a conclusion.

Hypothetical syllogism $P \rightarrow Q$ $Q \rightarrow R$ $P \rightarrow R$	Modus ponendo ponens $P \rightarrow Q$ P Q	Modus tolendo tollens $P \rightarrow Q$ $\neg Q$ $\neg P$
Disjunctive syllogism $P \lor Q$ $P \lor Q$ $\neg Q$ ili $\neg P$ P Q	Modus ponendo tollens $\neg(P \land Q)$ P $\neg Q$	Modus tolendo ponens $P \lor Q$ $\neg P$ Q

TABLE IV. DESIGN FOR LOGICAL FORM "CONCLUSION" – LEARNING DOMAIN "APPLICATION"

Learning domain: learning outcome

Application: To find logical statements and logical operators in a given text.

Task – solution

In displayed text it is necessary to find logical statements and logical operators.

„Oh, you can't help that“, said Cat: „we're all mad here. I'm mad. You're mad.“ Alice didn't think that proved it at all; however, she went on: „And how do you know that you're mad?“ „To begin with “, said the Cat, „a dog's not mad. You grant that?“ „I suppose so “, said Alice. „Well, then“, the Cat went on, „You see a dog growls when it's angry, and wags its tail when it's pleased. Now I growl when I'm pleased, and wag my tail when I'm angry. Therefore I'm mad.“ „I call it purring, not growling“, said Alice. „Call it what you like“, said the Cat.

Solution:
Logical statements: A dog's not mad.; A dog growls when it's angry, and wags its tail when it's pleased.; We're all mad here.; I'm mad.; You're mad.; I growl when I'm pleased, and wag my tail when I'm angry.
Logical operators: therefore, then, and, not

Content

In logic the statement is a set of terms by which something is claimed or denied. Statement can be simple or composed. Statement „*A girl named Alice is the main character in a fairy tale written by Lewis Carroll - Alice in Wonderland.* “ is an example of simple statement. Statement „ *A girl named Alice is the main character in a fairy tale written by Lewis Carroll - Alice in Wonderland,, and next to her in a fairy tale appear other interesting characters.*“ is an example of composed statement. Composed statement consists of two simple statements related by some link that represents an appropriate logical operator.
Displayed composed statement consist of two simple statements (*A girl named Alice is the main character in a fairy tale written by Lewis Carroll - Alice in Wonderland.* and *Next to Alice in a fairy tale appear other interesting characters.*) that are connected with „and“ which is a link for logical operator of conjunction.
Linkers and appropriate logical operators are:

1. negation – linkers: no, nor...
2. conjunction. – linkers: and, but, than the, already, although, though, however...
3. disjunction – linkers: or, either...
4. implication – linkers: if … than, only if...
5. equivalence – linker: if and only if.

TABLE V. DESIGN FOR LOGICAL FORM "CONCLUSION" – LEARNING DOMAIN "ANALYSIS"

Learning domain: learning outcome

Analysis: To extract all logical statements and logical operators from a set of logical statements, using which one can build a given form of conclusion.

Task – solution

From displayed set of logical statements it is needed to extract those logical statements and logical operators by which one can build conclusion in form of modus tolendo tollens.

A dog's not mad.
A dog growls when it's angry, and wags its tail when it's pleased.
We're all mad here. I'm mad. You're mad.
I growl when I'm pleased, and wag my tail when I'm angry.

Solution:
Logical statements: Dog's mad.; Dog growls when it's angry.; Wags its tail when it's pleased.
Logical operators: Not = ¬; (Only if) Then = →;
And = ∧

Content

It can be repeated content that is given in Tables 3 and 4.

TABLE VI. DESIGN FOR LOGICAL FORM "CONCLUSION" – LEARNING DOMAIN "SYNTHESIS"

Learning domain: learning outcome

Synthesis: To compile a logical form of conclusion by sequencing logical statements and link them by logical operators.

Task – solution

From given logical statements and logical operators should be compiled logical form of conclusion.

Logical statements: Dog's mad.; Dog growls when it's angry.; Wags its tail when it's pleased.
Logical operators: Not = ¬; (Only if) Then = →;
And = ∧

Solution:
Dog's not mad (only if) then (you see) a dog growls when it's angry, and wags its tail when it's pleased.

(Is such that) a dog growls when it's angry, and wags its tail when it's pleased.

Dog's not mad.

Content

It can be repeated content that is given in Table 3.

TABLE VII. DESIGN FOR LOGICAL FORM "CONCLUSION" – LEARNING DOMAIN "EVALUATION"

Learning domain: learning outcome

Evaluation: To check the validity of conclusion using a method of *reductio ad absurdum* and a boolean table.

Task – solution

It is needed to check the validity of given conclusion using a method of *reductio ad absurdum* and a boolean table.

(X ∧ Y) → ¬Z
¬¬Z

¬(X ∧ Y)

Solution:
Method of reductio ad absurdum

(X ∧ Y) → ¬Z	¬¬Z	¬(X ∧ Y)
~~t t t t f t~~	t f t	f t t t

Conclusion is valid because the absurdity is in the first premise. Namely, implication is not true if the first element (implicator) is true, and the second is false.

Method of boolean table

X	Y	Z	(X ∧ Y) → ¬Z	¬¬Z	¬(X ∧ Y)
t	t	t	t f f	t f	f t
t	t	f	t t t	f t	f t
t	f	t	f t f	t f	t f
t	f	f	f t t	f t	t f
f	t	t	f t f	t f	t f
f	t	n	f t t	f t	t f
f	f	t	f t f	t f	t f
f	f	f	f t t	f t	t f

Conclusion is valid, because there is no counterexample that would mean that from true premises follows the false conclusion.

Content

The validity of conclusion is one of the forms for preservation of truth. It can be checked by the method of *reductio ad absurdum* and by boolean tables. Both of these methods require translation of logical forms from natural language to the language of logic. Here we use a language of propositional logic. Language of logic has its own syntax and semantics. Syntax includes a list of symbols and formative rules, while semantics describes how formulas, occurred according to the rules of syntax, we can add the values.
Building of boolean tables for composed logical forms starts with determining the values of simple statements. These values are recorded on the left side of table. Every statement has value of true or false. By increasing the number of simple statements increases the combination of values on the left side of table (for example, for two numbers the combination is 2 * 2 = 4, for three it is 2 * 2 * 2 = 8, or in general for *n* is 2^n). Values of composed logical form are set from simple elements to more complex.
Method of reductio *ad absurdum* is a method in one line. Set of values go in reverse of set of values by boolean table – set of value takes place from values of main logical elements to those more simple. Both methods require knowledge of boolean tables of the main logical operations (negation, conjunction, disjunction, implication and equivalence).

IV. CONLUSION

This paper describes the first two phases (by ADDIE model) of the development of computer based system to support the acquisition of logical forms. The initial version of the analysis and part of the design is shown (for only one logical form of four of them). The system is based on problem-based learning and learner-content interaction.

The next phase in the development of the system is a further detailing of the design which should include the learning outcomes of all logical forms, definition of the ways of achieving verification of learning outcomes, and definition of levels of learning outcomes achievement which are considered acceptable.

Upon completion of the analysis and design phase follows the development phase. This phase should result in an initial version of the system with which learners will be able to be in online and offline interaction, and which will be widely available. For this purpose, the intention is to develop a system for the web environment (there is a possibility of interaction with other learners), but also as a standalone application for desktop, laptop and tablet computers, as well as smart phones.

In the implementation phase should be given an answer to the question of how the final version of the system will set up and how to make it available to learners. The intention is to set up a web version of the system on the server of the Polytechnic of Rijeka, which it already owns. On the same server will be set up a link through which the learner will be able to download the appropriate standalone application.

The evaluation phase should answer the question of how much the system successfully fulfils its purpose. To make this phase implemented, the intention is to put the system under evaluation of both, learners who have been interacted with the system, professors, as well as pedagogical experts.

REFERENCES

[1] B. B. Levin, "Introduction", in Energizing teacher education and professional development with problem-based learning, B. B. Levin, Ed. Alexandria, VA: Association for Supervision and Curriculum Development, 2001., pp. 1-7.

[2] C. E. Hmelo-Silver, "Problem-Based Learning: What and How Do Students Learn?", Educational Psychology Review, Vol. 16, No. 3, pp. 235-266, September 2004.

[3] D. R. Clark, "Bloom's Taxonomy of Learning Domains", http://www.nwlink.com/~donclark/hrd/bloom.html, 29.01.2014.

[4] J. J. Lecercle, Philosophy of Nonsense: The Intuitions of Victorian Nonsense Literature, London: Routledge, 1994.

[5] L. D. Spence, "Problem Based Learning: Lead to Learn, Learn to Lead.", http://n-imagecache.aldenhosting.com/~n1studyg//pblhand book.pdf, 29.01.2014.

[6] L. S. Cauman, First-order Logic: An Introduction, Berlin: Walter De Gruyter, 1998.,

[7] M. Savin-Baden, A Practical Guide to Problem-based Learning Online. New York: Routledge, 2007.

[8] P. Jones and R. Davis, Instructional Design Methods Integrating Instructional Technology, in Instructional Design: Concepts, Methodologies, Tools, and Applications, M. Khosrow-Pour, Ed. New York: Information Science Reference, 2011., pp. 101-113.

[9] R. M. Branch, Instructional Design: The ADDIE Approach. New York: Springer, 2009.

[10] S. Kovač, Logika, Zagreb: Hrvatska sveučilišna naklada, 2006.

[11] S. Kovač and B. Žarnić, Logička pitanja i postupci, Zagreb: Kruzak, 2008.