УДК 811.111’322+811.111’42

Iryna Biskub

DIAGNOSTICAL CATEGORIZATION IN SOFTWARE DISCOURSE

The article presents a new approach to treating the concept of cognitive categorization in modern linguistics. One of the key questions cognitive linguistics has to provide answers to is the question how people cognate knowledge in the process of learning. Any learning procedure is based on acquiring knowledge provided in the verbal form. The optimal way of presenting the verbal knowledge is to provide definitions of the main concepts related to the domain. The present article offers an approach according to which it is possible to discriminate between characteristic and unique feature of the concepts while giving their verbal definitions. The research was carried out on the material of the Engilsh software discourse, i.e., the set of language means used in defining functions in graphical user’s interfaces (GUIs). One of the ways to experimentally address this question consists in determining whether some entity does or does not belong to a specified category on the basis of a description of the entity. This is what users normally do when they read the texts presented in GUI dialogue boxes.

Key words and phrases: categorization, concept, knowledge, definition, cognition.

Formulation of the research problem and its significance. One of the main issues in modern cognitive science is whether concepts have defining properties or not [1; 3; 5]. The notion of a verbal definition of an object appears to be of primary importance in view of establishing categorial potential of concepts [3, p. 51]. Apparently, the only way to research concepts is to provide their verbal definitions, and to pick up language means – verbalizers – to give names to the key notions within one particular concept [1].

Any learning procedure or the process of knowledge acquisition presupposes the implementation of a deductive verbal reasoning [3, p. 52]. One of the most effective instruments of a deductive verbal reasoning is the so called “concept-thesis model” of knowledge representation.

The goal and the specific tasks of the article. The purpose of this article can be defined as an attempt to analyze the mechanisms of how to use the concept-thesis model in providing relevant descriptions of the functions in modern English graphical user interfaces (GUI).Graphical user interface (GUI) is a type of interface that allowsusers to interact with electronic devices through graphical icons and visualindicators such as secondary notation. Sometimes GUI is opposed to text-based interfaces, where text is the only means of communication between a user and a computer [3]. But, in the predominant majority of cases modern software applications use both texts and graphical icons in their interfaces, so the term “GUI” is now universally used to denote user’s interfaces.

A series of elements conforming a visual language have evolved to represent information stored in computers. By “visual language” software developers mean language means used the in descriptions (definitions) of the software functions [3, p. 52], as well as other graphical devices (icons, pictograms, etc) to specify the functional potential of a program. This makes it easier for people with few computer skills to work with and use computer software. The most common combination of such elements in GUIs is the WIMP (“window, icon, menu, pointing device”) paradigm, especially in personal computers.

The WIMP style of human-computer interaction uses a virtual input device to control the position of a pointer, most often a mouse, and presents information organized in windows and represented with icons. The important innovation of modern GUIs is the use of the dialog boxes to provide verbal definitions of menu features once the user brings the cursor (or another pointing device) to the name of the function in the menu bar. Apparently, the main task to be resolved by GUI designers is to provide the user with the relevant description of the menu function.

The best way to do it is to use the traditional format of a definition, which is the most productive tool used in learning and knowledge acquisition. A definition is naturally based on the verbal description of the most essential features that characterize a particular software function.

Presentation of the basic content of the research and an interpretation of the results which were obtained. One of the ways to experimentally address this question consists in determining whether some entity does or does not belong to a specified category on the basis of a description of the entity. In the majority of cases, the description of a function in a GUI is presented in the so-called dialog box, and it contains in formation that is deemed to define the category. In some cases, the descriptions lack such information. The question of how much information the description should contain, as well as the manner of presenting this information involving graphical means is still in crying need of answer for software developers.

It should be mentioned that the amount of information in the descriptions varies dramatically. These two factors, i.e., description length and description type, provide a fairly strong rational efor determining whether the key concepts that characterize software functions are well defined or not. This discrimination between quantity and quality of information presented in verbal definitions, like much research on categorization, is rooted in the classical view of concepts. According to the classical view of concepts, the mental representation of a category consists of properties that are common and unique to the objects in the category [1, p. 3]. As an example, the conceptual representation of the category labeled *square* would involve properties such as having four sides of equal length and four fight angles. Since all squares are assummed to possess these properties, a figure lacking anyone of these properties cannot be a member of the category *square.* Each of the above properties is therefore said to be individually necessary for category membership. Since squares are the only figures to have all of the above properties, any figure with these properties must belong to the category *square.* The above properties are therefore jointly sufficient for category membership. Set so find ividually necessary and jointly sufficient properties are considered defining because they allow one to determine category membership in an all-or-none fashion.

Another perspective of using concepts in definitions is obtained from a probabilistic view of conceptual representation. According to this view, the mental representation of categories involves properties that are common to some but not necessarily all members of a category. Such characteristic properties are also assumed to be involved in the categorization process. In some probabilistic models the existence of necessary and defining properties is denied, such a strong stance is not essential [3].

What is really important here is that all types of properties, whether necessary, defining, or merely characteristic, be understood and processed in the same way. Within this perspective a special status for defining properties is being denied. At the same time the probabilistic perspective emphasizes on the fact that some properties are more diagnostic than others and are therefore weighed more heavily in categorical decisions.

The probabilistic perspective presupposes that categorization should therefore depend on the number and the predictable diagnosticity of the properties asserted or negated with respect to an object. The most important thing about the predictions derived from a probabilistic viewis the fact that they are predominantly based on user's confidence. Amazingly enough, confidence in categorical decisions can now be graded. The degree of user's confidence increases with the number of properties that are asserted about the entity to be categorized and decreases with the number of properties that are negated. It should be mentioned here, that properties that are common and unique to members of a category are more diagnostic of category membership and presumably weigh more heavily in categorical decisions than properties that are merely characteristic.

We have analysed and compared the definitions of the names of functions in the Microsoft Word 2007 in order to single out the properties that are common and unique to the category, and established their metaphorical uses in the software discourse. Any metaphorical mapping is based on imposing the properties of the source onto the target. Thus, category definitions appeared to specify these properties in the relevant way, so that the users of GUSs would be able to understand the meaning of software function.

The following table presents the comparative analysis of the dictionary definitions for some software menu functions (Microsoft Word 2007) and their definitions in GUI:

Table 1

Dictionary definition vs. GIU definitions (Microsoft Word 2007)

Category	Dictionary definition	GUI definition
Header (in Header and Footer)	*1. One that fits a head on an object.* *2. Information at the top of a page, especially things such as numbers that appear on each page of a document.* *3. One that removes a head from an object, especially a machine that reaps the heads of grain and passes the min to a wagon or receptacle.*	*In a disk or tape file, a set of data that resides permanently at the beginning.* *It may be used for identification only (type of file, date of last update, etc.), or it may describe the structural layout of the contents, as is common with many document and data base formats.*
Footer (in Header and Footer)	*1.One that is an indicated number of feet in height or length. Often used in combination.* *2.Textual information, such as a title, date, or page number, which appears at the bottom of each page of a document which is printed by a computer.* *3.A note placed at the bottom of a page of a book or manuscript that gives more information about something.*	*In a document or report, common text that appears at the bottom of every page.* *It usually contains the page number.*

Apparently, the metaphorical uses of the following names of the functions are based on the categorical properties of the following concepts: HEAD and FOOT:

Head – top of body [countable] the top part of your body that has your face at the front and is supported by your neck (LDOCE online);

Foot – plural feet [countable] body part, the part of your body that you stand on and walk on (LDOCE online).

The characteristic properties of the concepts HEAD and FOOT presented in the dictionary definitions are *top of body* for HEAD, and *stand on* for FEET. These properties are not reflected in software functions. Whereas, the systemic properties that are reflected in GUI are as follows: UP for HEAD, and DOWN for FOOT. They perform the diagnostical functions in the metaphorization process [4, p.84] and are, consequently, more categorically valuable than just characteristic properties shared by all member of the categories HEAD and FOOT.

Some definitions of the functions in software discourse are still based on emphasizing direct characteristic properties shared by all members of the semantic category. This is typically done by using synonymic words or expressions while defining functions in the dialogue boxes, e.g.:

Select Recipients – Choose the list of people you intend to send the letter to (Select→ Choose);

Edit Recipient List – Make changes to the list of recipients and decide which of them should receive your letter (Edit→ Make changes);

Finish and Merge – Complete the mail merge (Finish→ Complete);

Such rephrazing based on specifying characteristic features of the concept that is being defined, is commonly used for providing descriptive definitions to the concepts denoting particular functions that are unique for the software discourse, and are not relevant for the other communicative contexts, e.g.:

Insert Hyperlink – Create a link to a Web page, a picture, an e-mail address, or a program (Insert → Create);

Manage Sources – View the list of all the sources cited in this document (Manage → View);

Protect Document – Restrict how people can access the document (Protect → Restrict);

Arrange All – Tile all open program windows side-by-side on the screen (Arrange → Tile).

The so-called diagnostical properties mentioned above are effectively used in the definitions of the software functions based on metaphorical mapping [2, p. 202]. The diagnostical mapping is presented in Table 2.

Table 2

Metaphorical mapping in the definitions of the software functions

Function	Definition	Metaphorical mapping
Strike through	*Draw a line through the middle of the selected text.*	*Through→middle*
Subscript	*Create a small letters below the text baseline.*	*Sub→below*
Superscript	*Create small letters above the line of text.*	*Super→above*
Shading	*Color the background behind the selected text or paragraph*	*Shade→background behind*
Word Art	*Insert decorative text in your document.*	*Art→decorate*
Watermark	*Insert ghosted text behind the content of the page.*	*Water→ghosted*

As the examples in the table suggest, metaphorical mapping in the definitions of the concepts is not based on sharing the fundamental features of the semantic category. The process of recognizing the functional potential of GUI menu items is directed by sharing and distributing the diagnostical features of the concepts. In case with metaphors, these features are neither unique nor characteristic to the general category. They mostly specify the one of the most vivid conceptual characteristics, which due to their metaphorical basis appear to be easy to recognize and facilitate the process of understanding.

Conclusions and prospects for further research. In conclusion, we assume that verbal definitions of concepts are based on specific categorization procedures that focus on different aspect of categorical meanings. It is a unique feature of the software discourse to use various categorization mechanisms while defining the functions of GUIs in the dialogue boxes. The definitions provided in the dialogue boxes very rarely emphasize on the properties that are thought to be common to all members of a category. In general categorization procedures they may weigh more heavily in categorical judgments, but in software discourse unique feature of the concepts appear to be more valuable in providing the users with the meaning of the software functions. The use unique feature is especially productive in metaphorical mapping where the diagnostical potential of cognitive categorization is of primary importance. Such diagnostical properties cause a steeper change in confidence level of software users with the number of properties negated, but this categorization strategy is really helpful as a basis of human-computer interaction via GUI.

References

1. Lakoff G. Women, Fire and Dangerous Things : What Categories Reveal About the Mind / G. Lakoff. – Chicago and London : The University of Chicago Press, 1987. – 614 p.

2. Lakoff G. The Contemporary Theory of Metaphor / G. Lakoff // Metaphor and Thought / A. Ortony (ed.). – 2nd edition. – Cambridge : CUP, 1993. – P. 202–251.

3. Pylyshyn Z. W. Computing in cognitive science / Z. W. Pylyshyn // Foundations of cognitive science / M. I. Posnered. – Cambr. (Mass.) ; L. : MIT, 1989. – P. 51–91.

4. Searle J. R. Metaphor / Searle J. R. // Metaphor and Thought / A. Ortony (ed.). – 2nd edition. – Cambridge : CUP, 1993. – P. 83–111.

5. Taylor J. R. Linguistic Categorization: Prototypes in Linguistic Theory / J. R. Taylor. – 3rd edition. – Oxford : OUP, 2003. – 308 p.

Біскуб Ірина. Діагностична категоризація в дискурсі програмного забезпечення. У статті запропоновано новий підхід до розуміння феномену когнітивної категоризації в сучасній когнітивній лінгвістиці. На матеріалі англомовного дискурсу програмного забезпечення встановлено особливості діагностичної категоризації, яка сприяє розумінню функціонального потенціалу прикладних програм через прочитання їх визначень у діалогових вікнах, що надає графічний інтерфейс. Виявлено відмінності між категоризацією за характерними ознаками й категоризацією за унікальними ознаками, яку визначаємо як діагностичну категоризацію. Доведено, що в сучасному англомовному дискурсі програмного забезпечення саме діагностичну категоризацію покладено в основу метафоричного мапування, завдяки якому користувач опановує функціональний потенціал комп'ютерної програми через взаємодію з її графічним інтерфейсом. Діагностичну категоризацію вважаємо також когнітивною стратегією, основне завдання якої – відокремлення несуттєвих категоризаційних ознак під час ідентифікації функцій програмного забезпечення. Це сприяє оптимізованому використанню графічних ітерфейсів комп'ютерних програм як інструменту ведення діалогу користувача й комп'ютерної системи.

Ключові слова: категоризація, концепт, знання, визначення, пізнання.

Бискуб Ирина. Диагностическая категоризация в дискурсе программного обеспечения. В статье предложен новый подход к пониманию феномена когнитивной категоризации в современной когнитивной лингвистике. На материале современного англоязычного дискурса программного обеспечения были установлены особенности диагностической категоризации, которая содействует пониманию функционального потенциала прикладных программ путем прочтения определений в диалоговых окнах, которые предлагаются графическим интерфейсом. Предложены различия между категоризацией по характерным признакам и категоризацией по уникальным признакам, определенной нами как диагностическая категоризация. Доказано, что именно диагностическая категоризация лежит в основе метафорического перенесения значения в дискурсе программного обеспечения, благодаря чему пользователь осваивает функциональный потенциал компьютерной программы, взаимодействуя с ее графическим интерфейсом. Диагностическую категоризацию считаем также когнитивной стратегией, главным заданием которой есть отделение несущественных категоризационных черт во время идентификации функций програмного обеспечения. Это спрособствует оптимизированному использованию графических интерфейсов компьютерных программ как инструмента ведения диалога пользователя и компьютерной системы.

Ключевые слова: категоризация, концепт, знания, определение, познание.

УДК 81' 371

Lyudmyla Bulatetska

FROM COMMUNICATIVE STRATEGIES TO PRAGMATIC MOTIVATIONS

The article deals with the notion of involvement, which has the dynamic nature and varies within the boundaries of empathy, sympathy, (self)withdrawal, antipathy. Involvement is defined as a cognitive notion with the prevailing characteristics of cognitive distancing or approximation in interaction. Generally recognized is the point of view that involvement is a gradual phenomenon. It fluctuates depending on the activity of interlocutors. What follows from it is the absolute mutual trust of communicative partners to one another on the one hand or giving up and disregard the communication by both sides on the other. Such a shift of communicative parameters of communicative activity is a direct reflection of the change in the focus of orientation which directly depends on the focus of interest. The general vision of involvement is widening and depends on what goes to the focus of orientation. Existing approaches are reduced to the following three: a) involvement of the speaker and listener to the information of the topic (as individual participants); b) involvement of the community members (as mass participants); c) involvement of the topic on the parameter of its informativity and validity of the problem. Degrees of involvement are controlled by the communicative strategies of interlocutors, they re-frame the configuration of notions of empathy, sympathy, (self)withdrawal and antipathy, assisting in achievement of maximal communicative result and coincidence of positions of opposite sides.

Key words: involvement, intention, focus of orientation, vantage point, empathy, sympathy, (self)withdrawal, antipathy, prediction, anticipation, dynamic equilibrium.

Formulation of the research problem and its significance. Involvement as a communicative entity is determined by the degree of mutual interaction of interlocutors directed towards effective