

Terminological Abstractions for Terminology Classification

Igor Boyko, Ph.D.
HP Labs, Palo Alto

Abstract

The paper considers the method of technical terminology classification on the basis of terminological abstractions. The method is used to overcome jargon differences between scientific domains. Each scientific domain has its own terminology, or jargon. Within a domain, this is valuable as it provides conciseness and exactness. Outside the domain, however, it results in confusion. The experts within the domain can understand the same term in its different senses; to those outside the domain, it is unclear. The development of semantic classifiers of abstract terminology as intermediate links between various domains will help to solve this problem. This classifier may also serve as a form of ontology for the applicable domains. Such classifiers unite linguistic-semantic and functional approaches for analyzing the terms in a natural language. The linguistic approach uses a natural language lexicon and considers the terms as parts of speech: e.g. as verbs, nouns, or adjectives. In the functional approach we consider a verb as an action, a noun as a subject or an object, and an adjective as an attribute of: the action, the subject or the object.

Universal Semantic Code (USC) offers a method of semantic representation of terms using the verb as an action. USC has its own set of formal tools, called the USC algebra. We apply the USC approach to the terminology where a noun is the subject or the object, and an adjective is the attribute of: the action, the subject or the object. We develop an algorithmic approach for processing meta-terminology or meta-data. The formal tools support the meta-representation of the terminology to develop the multi-language classification with the collection of terms on a functional basis. We propose a system for terminology standardization by the functional criteria with the relevant relations inside of the classes.

Introduction

The terminology classification is a main point that determines the meaning of terms of scientific domains. There is important for users of computer-applied systems to be sure that the computer can process a terminological input to give a relevant output. The ideal way for such communication is communication on the same terminological language (Scragg, 1978).

The traditional approach to the terminology classification is the classification on the object basis. For example, chemistry classifies substances in correspondence with a number of atoms within a molecule, physics gives the class of the elementary particles and biology considers the classification of the plants. The technical field proposes classifications of technical systems on the object basis: vehicle, building material or weapon.

The words: substance, elementary particle, plant, vehicle, building material, and weapon represent general or abstract notions. Modern languages of the object-oriented programming have the similar structure for representation of classes and objects of the classes.

Universal Network Language (UNL) proposes interesting approach to the word processing and classification. UNL is the intermediate language for the Internet on the basis of logical expressions (Uchida and others, 1999). Unfortunately, the language uses a traditional approach to the classification of words.

The most interesting classification is the functional classification (Martynov and others, 1992), because the formal tools support it.

USC provides substitution of real words by the USC formulas (Martynov, 2002). The formulas represent the meaning of real words and can be introduced in the UNL instead of real words or used independently for knowledge representation.

Since USC declares functional approach to the world we do not consider chemical or physical terminology as the functional terminology and consider technical terminology as the functional terminology and the technical world as the functional world. Our main interest is: What the action the object performs? (Boyko, 1996, Martynov, 1996).

The kind of the natural language does not influence on the functional terminology classification because the formal interpretation of terms has the abstract representation.

The functional classification and the formal representation of terminology are applied for development of the natural language processing (NLP) computer systems, for example: summarizers, categorizers, semantic searchers or machine translators. Moreover, the formal representation of terms of the technical world helps in solving of engineering problems and to find inventive solutions of the problems (Boyko, 2001).

Functional Classification of Technical Actions

USC is a semantic representation language that has been developed for a long time from a version to version. The newest version is the USC-6 (Martynov, 2002).

The USC algebra operates own formulas and has a notation:

$$\mathbf{A} = \langle \mathbf{M}, \rightarrow, ' \rangle$$

Where:

$\mathbf{M}$ - is the set of elements;

$[\rightarrow]$ - is the binary operation on the given set of elements (operation of implication);

$[']$ - is the unary operation on the given set of elements (operation of negation).

The main unit of the USC representation is a complex formula that represents the interaction of variables of the formula. The semantic relations in the complex formulas have a natural language interpretation:

$((X \rightarrow Y) \rightarrow Z) \rightarrow ((Z \rightarrow W) \rightarrow Y')$ - X by means of Y acts on Z so Y preserves Z from W

or for simplicity:

$((XY)Z)((ZW)Y')$ -- X by means of Y preserves Z from W.

The variables of the formula must be substituted with specific data. For example, a man (X) by means of a metal can (Y) preserves food (Z) from germs (W).

We would like to draw attention to the fact that the left part of the formula causes the potential action in the right part. Thus the complex formula reflects the following situation: Stimulus → Reaction.

Each complex USC formula has one and only one meaning. Each conversion of the USC formula into another has one and only one meaning conversion. The USC axioms include the rules of conversion. The USC formulas represent all possible abstract technical actions.

The USC-classifier comprises all technical actions paired with their USC formulas. In the USC classifier all technical actions are the functional analogues, where:

- each abstract technical action correlates with own USC formula
- each USC formula determines some technical action and has verbal interpretation
- each action-analogue correlates with the abstract action

The classifier helps to avoid the linguistic problem of polysemy because one action-analogue can be included in different classes.

For the first glance it seems similar to the dictionary of synonyms but it is not correct because all actions are classified as the functional analogues (Fig.1, Fig.2).

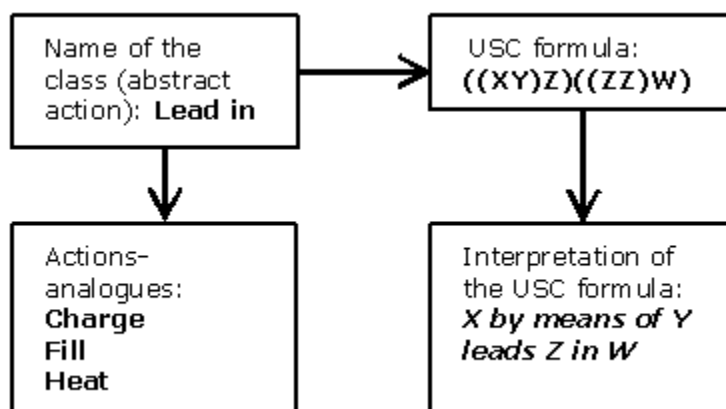


Fig. 1. The structure of the USC classifier for the technical action lead in

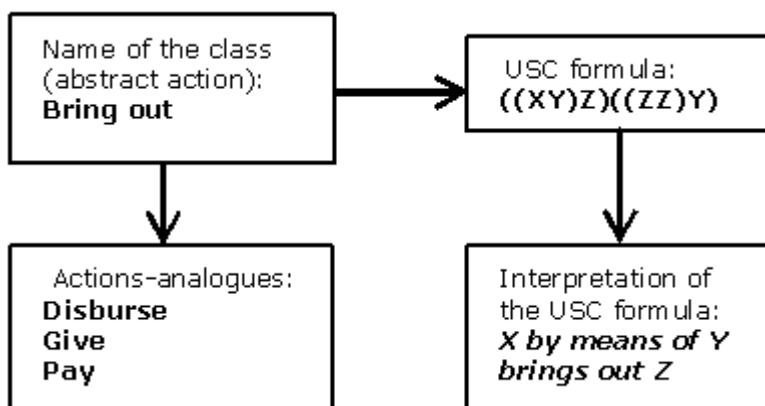


Fig. 2. The structure of the USC classifier for the technical action give

Actually an abstract USC action is a meta-term for representation of the class of actions. The classifier is a non-traditional system of technical actions-analogues. The classifier is built according to the function that each action-analogue performs.

The USC formulas have the single sense, the strict length, and the limited set of the variables. The USC classifier of technical actions consists of 96 functional classes.

Multilingual Representation of Actions

The formulas of USC represent the actions of any natural language. For example, in English the USC formula: ((XY)Z)((ZZ)W) represents the action-analogue 'heat'.

In French the word 'chauffer', in German the word 'anheizen', and in Italian the word 'calore' have the meaning 'heat' and can be represented by the same USC formula. In this case the USC formula is the intermediate code for representation of the meaning 'heat'.

French word 'chauffer' has own list of actions-analogues of French language, for example: rechauffer, chaleur, feu, ardeur.

German word 'anheizen' has own list of actions-analogues of German language, for example: anwarmen, aufheizen, beheizen, erhitzen.

Italian word 'calore' has own list of actions-analogues of Italian language, for example: ardore, riscaldamento, prova eliminatoria.

Each of the term can be represented by the same USC formula.

Another example of the USC formula: ((XY)Z)((ZZ)Y) represents the action-analogue 'cool'.

In French the word 'rafraochir', in German the word 'kälte', and in Italian the word 'freddo' have the same meaning 'cool' and can be represented by the same USC formula. In this case, similar to the previous example, the USC formula is the intermediate code for translating of the words from one language into another.

French word 'rafraochir' has own list of actions-analogues: refroidissement, froideur.

German word 'kälte' has own list of actions-analogues: kühle, kaltstellen.

Italian word 'freddo' has own list of actions-analogues: infreddolimento, gelo.

Obviously any of the terms can be represented by the same USC formula.

Thus the database of technical actions comprises set of the USC formulas for representation of any kind of technical actions both the abstract and specific. Such representation does not depend on a kind of a natural language and allows storing the word having different meanings in the different classes.

Classification of Objects

The USC formulas represent both actions and objects.

We distinguish two types of objects:

1. Non-functional
2. Functional

A lot of object classifications exist where the non-functional objects are classified on the traditional basis: substance, field, etc.

We classify functional objects in correspondence with the function that must be performed. In other words, if objects: heater, calefactor, and warmer perform the main function heat then they are members of the class heater and have the formal representation similar to the action heat.

The objects: cooler, refrigerator, and cryostat perform the main function cool. Therefore, they are members of the object class cooler and have the formal representation similar to the action cool.

The only mark that distinguishes the action and object formulas is the character O with the object formula. For example, the formula of heater:

$O((XY)Z)((ZZ)W)$

and the formula of cooler:

$O((XY)Z)((ZZ)Y).$

It is very important for the computer systems to manipulate with the knowledge that have the similar representation and strict semantic relations (Fig. 3).

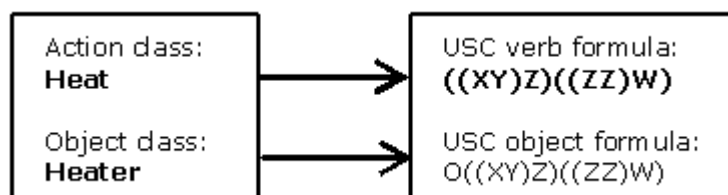


Fig. 3. Relations of the formulas of the technical actions and objects

The USC classifier classifies technical objects in 96 functional classes similar to 96 functional classes of the technical actions.

Multilingual Representation of Objects

The formulas of USC represent functional objects of any natural language similar to technical actions.

In French the word 'radiateur', in German the word 'erwarmer', and in Italian the word 'stufa' have the meaning 'heater' and can be represented by the same USC formula. The USC formula is an intermediate code for translation from one language into another.

French word 'chauffer' has own list of actions-analogues: appareil de chauffage, réchauffer.

German word 'erwarmer' has own list of actions-analogues: heizer, hitzdraht.

Italian word 'stufa' has own list of actions-analogues: riscaldamento, termosifone.

Thus the database of the technical objects comprises set of the USC formulas for representation of technical objects both abstract and specific. Such representation does not depend on a kind of a natural language and allows storing the word having different functional object meanings in the different classes.

Classification of Attributes

Similar to the classification of technical actions and technical objects the USC formulas represent attributes of actions or objects.

We distinguish two types of attributes:

1. Non-functional
2. Functional

Non-functional attributes can be classified on the basis of non-functional objects. For example, from the non-functional object wood the attribute is wooden, from the non-functional object dust the attribute is dusty etc.

Functional attributes are classified by the actions. In other words, if the action is heat then the attribute is hot, if the action is cool then the attribute is cold etc. Functional attributes are members of classes of attributes and have the formal representation like the actions heat or cool.

The only mark that distinguishes the formulas of the attributes is the character A with the formula. For example, the formula of the attribute hot:

$$A((XY)Z)((ZZ)W)$$

and the formula of the attribute cold:

$$A((XY)Z)((ZZ)Y).$$

This is a very strong point of complicated computer systems to have the universal semantic representation of knowledge and strict semantic relations (Fig. 4).

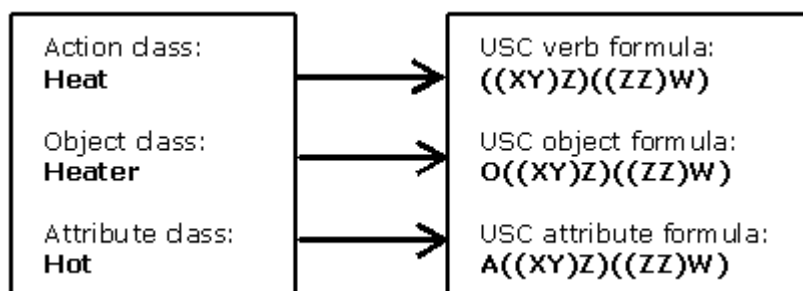


Fig. 4. Relations of the formulas of the technical actions, objects and attributes

The USC classifier classifies technical attributes for 96 functional classes similar to the 96 functional classes of technical actions and objects.

Multilingual Representation of Attributes

Thus the database of attributes comprises the set of the USC formulas for representation of technical attributes both abstract and specific. Such representation does not depend on a kind of a natural language.

In French the word 'chaud', in German the word 'heiß', and in Italian the word 'caldo' have the meaning 'hot' and can be represented by the same USC formula. The USC formula is an intermediate code for translation of attributes from one language into another.

French word 'chaud' has own list of actions-analogues: échauffé, excité.

German word 'heiß' has own list of actions-analogues: scharfgewürzt, warm, scharf.

Italian word 'caldo' has own list of actions-analogues: calda, bollente.

Thus the database of attributes comprises the set of the USC formulas for representation of technical attributes both an abstract and specific. Such representation does not depend on a kind of a natural language and allows storing the attributive word having different meanings in the different classes.

Conclusion

The USC classification of technical actions, objects, and attributes is a classification of technical terminology on the functional or abstract level.

The most abstract terms are represented as classes of terms. Each abstract term is specified by terms of the particular domain. The specific terms are included in the lists of actions-analogues of the class.

Imagine the possibility of communication of two professionals of two scientific domains by means of abstract terminology. Such communication must be much easier for mutual understanding of professionals, for example, of chemistry and astronomy.

The specialized chemical publications for chemists are unclear for astronomers and specialized astronomical publications are unclear for a chemist. However, both chemical and astronomical phenomenon may be described by abstract terms.

Using of abstract terminology initiates not only comfortable mutual understanding but gives a possibility to find homogeneity of sciences.

Abstract terminology helps to invent new technologies (Broadbent, 1966). The formal representation of technological processes helps in computation of inventive solutions (Boyko, 2001).

In the NLP field the semantic representation of abstract terminology allows to develop new methods of semantic search of information within databases of local computer, Intranet or Internet. Besides, the semantic representation may be used for development of systems of text summarization, text categorization, texts correlation, etc.

The semantic theories similar to USC facilitate structuring both technical information and non-technical information and establish semantic relations between them (Gordey, 1998).

References

Boyko, I.M. 1996. Semantic processor. Invention Machine Project-96. Boston. Cambridge.

Boyko, I.M. 2001. Computer Semantic Search of Inventive Solutions. TRIZ Journal. USA.

Broadbent, G.H. 1966. Creativity. The design method. London.

Gordey, A.N. 1998. Computation principles of semantics of subject domains. Belarus State Linguistic University. Minsk.

Jackson. P. 1999. Introduction to expert systems. Addison Wesley Longman Limited. England.

Martynov, V.V., Boyko, I.M., Gyminski, A.P. 1992. Knowledge Bases Construction of Systems for Solving Intellectual Problems. Controlling Systems and Machines. 5/6.

Martynov, V.V. 1996. USC calculus of key words and key ideas. Invention Machine Project96. Boston. Cambridge.

Martynov, V.V. 2001. Foundations of Semantic Coding. Summary. European Humanity University. Minsk.

Skragg, G. 1978. Semantic nets as a memory models. Computational Semantics.

Uchida, H., Zhu, M., Della Senta, T. 1999. A gift for millennium. Institute of Advanced Studies. The United Nations University. Tokyo. Japan.