The Public Journal of Semiotics II(2), July 2008, pp. 30-51

Conceptual blending and sign formation

Hubert Kowalewski
UMCS, Instytut Anglistyki
pl. Marii Curie-Skłodowskiej 4
20-031 Lublin, Poland
almunafikun@gmail.com

Abstract

In this article I will investigate the process of conceptual blending involved in sign formation. The main objective of this article is to demonstrate that conceptual blending theory is capable of accounting for the creation of both linguistic and non-linguistic signs from pre-existing semiotic inventory. Moreover, like in the case of logos and names of certain products, the conceptual mechanism behind the formation of linguistic and non-linguistic signs is similar not only in general aspects, but also in fine-grained details. This statement is by no means paradoxical. The theory of conceptual blending strives to describe the basic conceptual mechanism responsible for the semiotic capabilities of the human mind and is not intrinsically connected with any specific type of signs; thus, cognitive strategies which prove to be effective for the creation of, for instance, graphic signs may be reused for the creation of linguistic signs.

Keywords: logos, blending, cognitive

1 Introduction

In this article we will examine the process involved in the creation of novel signs from pre-existing semiotic inventory. This type of process is not uncommon; few signs are formed in complete isolation from other signs or are devoid of any motivated relationship with existing structures. The main thesis of this article may seem paradoxical – my intention is to demonstrate that in some cases signs of different types, like linguistic and graphical ones, may be created in similar way and that there are significant parallelisms in the conceptual mechanism underlying formation of these types of signs. These parallelisms can be described in a systematic and methodologically coherent manner. Material analyzed in this article consists of logos and names of Linux based-computer operating systems and theoretical framework adopted for the purpose of the analysis is the theory of conceptual blending devised by Gilles Fauconnier and Mark Turner.

Before we proceed to the case studies, it may prove useful to have a quick look at Linux itself. "Linux" is a customary name for a vast array of computer operating systems or, to be more precise, the core components of an operating system known as **kernel**. The kernel is the most important "layer" of an operating system responsible for coordinating components of hardware. In principle, kernel is a complete, self-contained and functional operating system, but in practice its usability is limited due to the fact that it does not contain any graphic user interface (abbreviated as GUI), let alone any "desktop" applications. Thus, virtually all widely-used Linux operating systems have another layer, a more intuitive graphic user interface called **desktop environment**, and additional programs, like word processors, spreadsheets, etc.[1] A complete system, including a kernel, a user interface and additional applications is a **distribution**.

An important difference between Linux-based and other operating systems is the modularity of the former. Modularity is often explained via a building blocks metaphor; in principle, a distribution may be created from scratch through arbitrary assemblage of components. Thus, users may select any kernel, combined it with any desktop environment and any additional software they find necessary. Moreover, one distribution may be equipped with more than one desktop environment, which can be freely toggled. Nevertheless, building a customized distribution requires considerable knowledge and programming skill, therefore, for the sake of convenience, most distributions come pre-assembled with a GUI and most useful applications.[2]

One of such distribution is Ubuntu, an operating system with a Linux kernel and a desktop environment called GNOME. Ubuntu was designed to be user friendly, easy-to-use and aesthetically pleasing. The effort of the authors was appreciated by users and the distribution became a tremendous success – currently, it is one of the most widely used distributions in the world. The success was so great that the authors decided to launch another project aiming at creating an operating system with similar collection of applications, but equipped with an alternative desktop environment called KDE. This new distribution was called Kubuntu. Unfortunately, both distributions were rather demanding on hardware and not suitable for older machines. Hence, Ubuntu developers decided to release another Ubuntu derivative, Xubuntu, with a light-weight desktop environment called XFCE. Soon Ubuntu and its derivatives became so popular that independent programmers started to create other Ubuntu-based distributions, which were not officially supported by the authors of original system. These versions feature other desktop environments or additional software not included in official releases.

It is worth mentioning that the difference between various desktop environments lies not only in appearance, but, more importantly, in functionality. Despite of the fact that most of Ubuntu derivatives are equipped with a different desktop environment, they are based on the same core components, which makes them similar in many ways. From the semiotic point of view, this situation should be reflected both in names and logos accompanying every distribution. Indeed, as we will soon see, semiotic inventory associated with every distribution attempts to highlight both differences and similarities between various derivatives. This effect is achieved through harmonious composition of seemingly dissimilar elements of various signs. Moreover, the process responsible for composition of signs is essentially the same for both linguistic and non-linguistic signs (names

[1] Strictly speaking, the matter is slightly more complicated, as a typical desktop operating system consists of three layers – kernel, window manager and desktop environment – although in some cases it is difficult to clearly delineate the last two. In addition, the word “layer” is metaphorical, far from technical precision and would probably never be used by a software developer. Nevertheless, for the purpose of this article let us accept this slightly simplified view.

[2] Particular components of a distribution can be manipulated freely due to less restrictive copyright policy adopted by developers of Linux related software. Most of the elements, including the kernel, most of desktop environment and an overwhelming majority of the additional programs are released under open licenses. Software under such licenses are free of charge and may be freely modified and reused by anyone.

and logos). The process in question is best described within the theoretical framework of conceptual blending.

2 Conceptual blending

The conceptual blending (also known as blending theory – BT), proposed by Gilles Fauconnier and Mark Turner (1995), is probably most frequently employed in the study of language. In linguistics this theoretical framework is used extensively to account for a wide variety of phenomena including, among others, conceptual metaphor, metonymy (e.g. Turner and Fauconnier 2003) and discourse structure (e.g. Oakley 1998). Turner and Fauconnier, however, never claimed that their theory is applicable to linguistic data alone. On the contrary, on many occasions they used conceptual blending to analyze visual data (e.g. cartoons) and cultural images (e.g. the Grim Reaper, both in Fauconnier and Turner 2007). Therefore, it seems reasonable to claim that blending theory is not restricted to any particular type of data, as it grounded in general conceptual and cognitive mechanisms universal for all semiotic activity of the mind.

Conceptual blending makes use of so called **mental spaces**. The notion was proposed by Fauconnier (1985) and can be defined as "a partial and temporary representational structure which speakers construct when thinking or talking about perceived, imagined, past, present or future situation." (Grady, Oakley and Coulson 2007). Despite certain linguistic bias evident in the quotation, mental spaces should not be thought of as phenomena of language alone. Their nature is conceptual rather than linguistic; their usefulness in the study of language is the consequence of the fact that linguistic expressions are phonological manifestations of more general conceptual processes.

Conceptual blending involves (at least) four mental spaces. Two of them are **input spaces** containing semantic structures contributed to the "output" structure. The third space, called **generic space**, embraces a schematic structure shared by both inputs. The content of the generic space represents abstracted commonalities occurring across all input structures, regardless of how dissimilar these structures may appear. The shared structure is a kind of generalization derived at the expense of fine-grained details, which are temporarily suppressed or overridden. These commonalities become the foundation of a new, emergent structure. This novel structure appears in the last space involved in conceptual blending, i.e. **the blended space**. The process is rendered graphically in Fig. 1.

In the illustration the solid lines represent correspondences (i.e. conceptual associations between elements of semantic structures) across input spaces. Corresponding elements are mapped into the generic space giving rise to the shared structure. The blended space recruits the schematic structure of the generic space, giving rise to a more specific construction than fairly general structure of the generic space, as it is elaborated with details from the input spaces. Several points should be made here. Firstly, the blended space combines elements of input spaces, so that components of two separate spaces are "blended" into one structure. This does not necessarily mean that corresponding components are mapped into a single element. The emergent structure should be a coherent array of conceptualization, but it can feature several separate elements linked in a systematic manner, rather than a single monolith entity. If corresponding elements from the inputs are combined into one element, **fusion** takes place. Fusion is typical for metaphors, in which one entity "becomes" another entity (cf. a metaphorical expression *Time is money*). Secondly, the emergent structure (shown in Fig. 1 as a rectangle inside the blended space) may recruit elements of input spaces which are not mapped into the generic space and have no cross-space correspondences. Additional components may be recruited opportunistically if, for any reason, they are perceived as relevant for the emergent structure. Thirdly, the process of blending is not merely additive. Once the emergent structure is created, it is capable of attracting new semantic elements, absent from the inputs and the generic space. In Fig. 1 these elements are represented as dots "floating" freely inside the emergent structure.

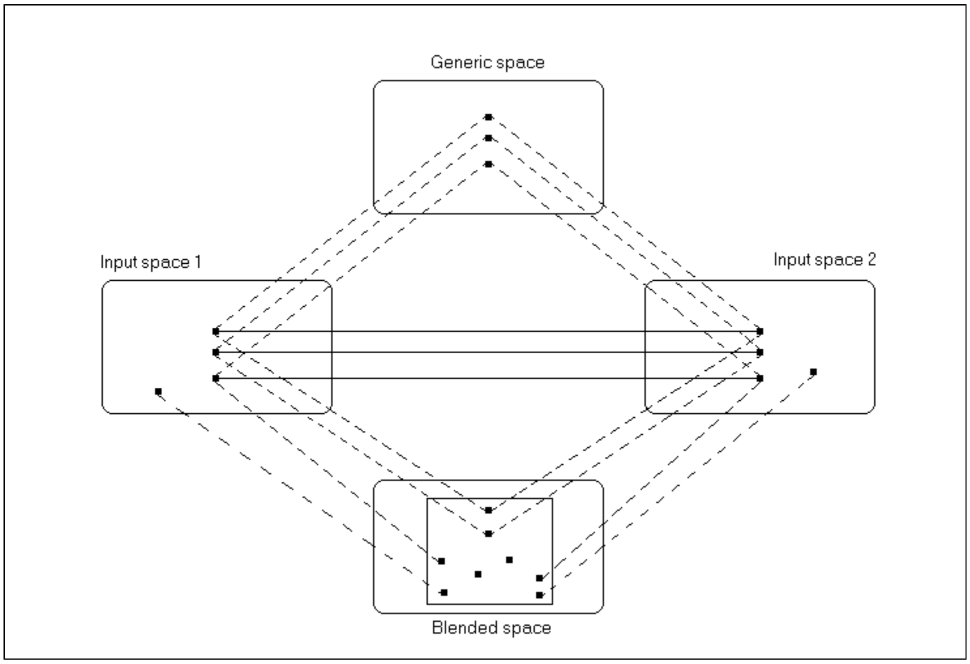

Figure 1.

Let us illustrate these three points with actual linguistic data. The following analysis of (1) proposed by Turner and Fauconnier (2003) has become a classical example of how BT can be used for explaining complexities of metaphorical expressions. The authors examine a metaphorical statement produced in 1998 after an alleged sexual scandal involving President Bill Clinton:

(1) If Clinton were the Titanic, the iceberg would sink.

In this case the conceptual blending features two input spaces: one containing the "Clinton scenario," in which the President survives damage caused by the alleged scandal, and the other including the "Titanic scenario," in which the ship sinks after colliding with an iceberg. The authors summarize the process of blending in the following passage:

> There is a partial cross-space mapping between these inputs: Clinton is the counterpart of the Titanic and the scandal is the counterpart of the iceberg.

> There is a blended space in which the Clinton is the Titanic and the scandal is the iceberg. This blend takes much of its organizing frame structure from the Titanic input space – it has a voyage by a ship toward a destination and it has the ship's running into something enormous in the water – but it takes crucial causal structure and event shape structure from the Clinton scenario – Clinton is not ruined but instead survives. There is a generic space that has structure taken to apply to both inputs: one entity that is involved in the activity and is motivated by some purpose encounters another entity that poses an extreme threat to that activity. In the generic space, the outcome of this encounter is not specified. (Turner & Fauconnier 2003: 133)

Further, the authors notice that some parts of the emergent scenario are not supplied by any of the input spaces (these are the "floating" dots in Fig. 1). Consider the result of the encounter in the emergent scenario, i.e. the iceberg sinking after collision with the ship. Obviously, this element is not contributed by the Titanic scenario; originally, it was the ship that sank, not the iceberg. The Clinton scenario does not provide an explanation either – the President does survive the scandal, but there is no semantic content representing the scandal being "damaged" in any way. This element does not come from the generic space, as the authors state clearly that here "the outcome of the encounter is not specified." What is more, the result of the encounter in impossible and contradicts the content of the "Titanic scenario" – we realize that icebergs do not sink after colliding with ships. Thus, it is legitimate to claim that this part of the scenario is an innovation provided by the emergent structure having no counterparts in other spaces. Novel elements are not uncommon in the blended space; emergent structures are sometimes governed by their own internal "logic," which may override the "logic" of the inputs.

The example demonstrates the already mentioned process of **fusion**. In (1) Clinton and the Titanic are fused into a single object (in the metaphorical understanding, Clinton *is* the Titanic) and so are other corresponding elements, i.e. the scandal and the iceberg. However, the blended space embraces much more than two object; in fact it features a whole scenario, which contains unfused elements as well. Consider the outcome of collision. This element is not a result of fusion, as it is not contributed by any of the inputs. Thus, fusion is not obligatory for creation of the emergent structure – elements contributed by inputs may be incorporated into the blend in a different manner.

Fauconnier and Turner (2007 [1998]) formulated several **optimality principles** governing the creation of conceptual blends. They are summarized neatly by Grady, Oakley and Coulson (2007: 425-426):

Integration:

The scenario in the blended space should be a well-integrated scene.

Web:

Tight connections between the blend and the inputs should be maintained, so that an event in one of the input spaces, for instance, is construed as implying a corresponding event in the blend.

Unpacking:

It should be easy to reconstruct the inputs and the network of connections, given the blend.

Topology:

Elements in the blend should participate in the same sorts of relations as their counterparts in the inputs.

Good reason:

If an element appears in the blend, it should have meaning.

Metonymic tightening:

Relations between elements from the same input should become as close as possible in the blend.

The optimality principles should not be thought of as rigid laws but must be treated with fair degree of flexibility. As the Grady, Oakley and Coulson state, “[there] is a tension among some of these principles, and so each blend satisfies them to varying degree.” (2007: 426)

3 Blending and logos

Let us proceed to the analysis proper. In this section we will have a look at logos of Ubuntu-based distributions and in the following section we will examine their names. The starting point is the Ubuntu logo presented in Fig. 2(a). As already mentioned, Ubuntu is a distribution equipped with GNOME desktop environment, whose logo is shown in Fig. 2(b). In this case the process of conceptual blending does not apply, as the Ubuntu logo does not include any elements of the GNOME logo. The reason for this is the fact that GNOME is a default desktop environment of Ubuntu and therefore there is no need of marking it explicitly. Since desktop environment is an important part of any distribution, we may assume that the knowledge about this component is a significant part of the notion of a particular operating system. Consequently, it is reasonable to conclude that notion of GNOME is an essential element of semantic structure representing Ubuntu. In other words, the notion evoked by Ubuntu logo contains implicit reference to GNOME, even though it does not appear in the signifier.

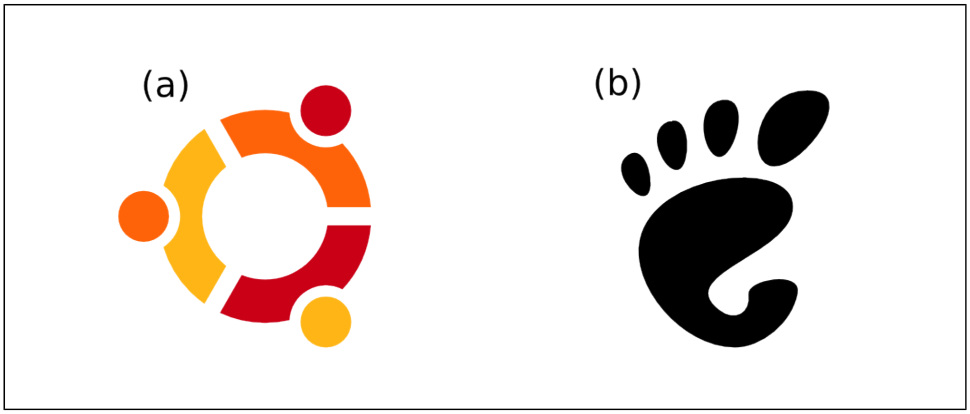

Figure 2. (a) Ubuntu logo; (b) GNOME logo.

The chronologically first derivative is Kubuntu equipped with desktop environment called KDE. Kubuntu and KDE logos are shown in Fig. 3(a) and 3(b) respectively. As we can see, Kubuntu logo is a blend of elements derived from Ubuntu and KDE logos. Input spaces contain graphic structures[3] of Ubuntu and KDE logos. The generic space embraces a schematic structure shared by both signs, in this case this is a round entity. In the Ubuntu space the shape corresponds to (either of) three little circles at the edges of the Ubuntu logo, in the KDE space it is linked to the overall shape of the gear. Abstracting the shared structure from the KDE space requires notable cognitive effort. Firstly, the gear, which is rendered only partially in the KDE logo must be conceptually completed into its full form. This is facilitated by a perceptual phenomenon called **gestalt perception**, in the result of which it is possible to mentally "fill in" parts of known structures which are not present in actual images. Gestalt perception is so pervasive and automated that it operates even when the logo is not considered in the context of conceptual blending. The most natural way of looking at the KDE logo is perceiving it as a combination of a complete gear and a letter, even though only part of the gear is rendered graphically. Certainly, it would be highly unusual to claim that the logo consists of a letter and a part of the gear or a broken gear, even though this is what the sign actually features. Secondly, the depiction of a gear is stripped of fine-grained details, as the generic space should contain a structure schematic enough to be shared by all inputs. In plain words, what is mapped into the generic space is not the concept of a gear as such, but a more general notion of a round structure, which the gear instantiates.

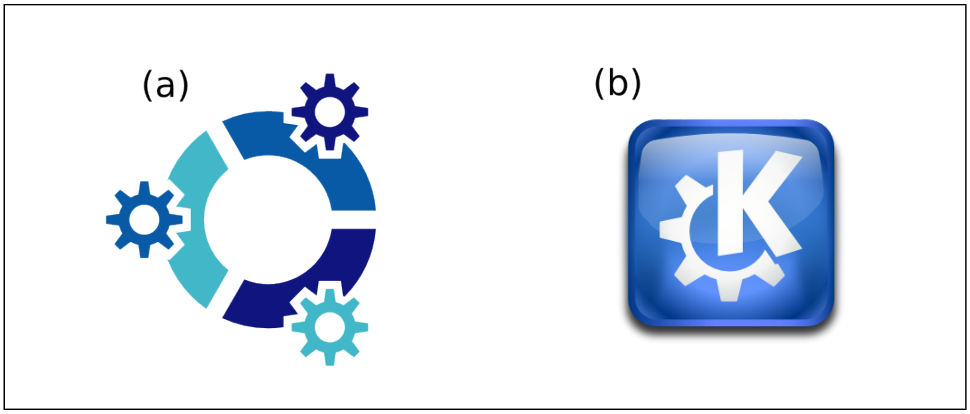

Figure 3. (a) Kbuntu logo; (b) KDE logo.

In the blended space elements from the inputs and the generic space are combined into a one structure (thus, fusion takes place). The round shape from the generic space is elaborated according to the cues provided by the KDE input into a gear and located at the edges of the "circle of three" recruited from the Ubuntu input. In this way, a novel sign combines recognizable elements of both logos. In order to provide additional cues of KDE, the blue color is mapped from the KDE

[3] The term "graphic structure" should be approached with caution. Conceptual blending, being a mental process, cannot involve any material entities. Therefore, through "graphic structure" I understand mental representations of real world entities constructed on the basis of sensory data rather than concrete markings on any kind of physical material. This provision should be borne in mind throughout the whole article.

input.[4] The presence of blue is not merely a fancy on the part of the artist (such a fancy would violate the optimality principle of good reason). As the overall shape of the novel sign is derived from the Ubuntu logo and the KDE components are rather small in comparison to the main circle-of-three, additional elements from the KDE input may be used to bring a conceptual balance to the novel sign. The resulting blend a coherent unified structure; Kubuntu logo is naturally perceived as a single entity, not a loosely knit combination of random elements. On the other hand, particular components of the blend are readily recognizable as recruited from relevant input spaces. These fact reflects the principles of integration and unpacking.

In principle, the process is quite similar in the case of Xubuntu, a light-weight distribution with XFCE desktop environment. The novel logo is a tightly knit combination of elements derived from input signs. The details of the process, however, are slightly different and deserve extended discussion. Fig. 4(b) depicts XFCE logo and 4(a) the logo of Xubuntu. This time it is far more difficult to extract the generic space structure – at first glance, two logos seem to share no elements from which the generic structure may be recruited. As a consequence, the novel sign may seem to be a somewhat random combination of arbitrarily selected components. Nevertheless, after closer inspection, it appears that both input signs share a similar pattern of general spacial organization; more specifically, particular elements of both signs are laid out according to center-periphery scheme. In Ubuntu logo parts of the circle of three are located on the perimeter of a circle around a central point (specific, though not marked graphically). In XFCE logo the background X shape marks the central point of the sign in the place, where the two arms cross. This location is also occupied by the image of a mouse. Consequently, this central point is visually most prominent and the whole image "gravitates" towards the center. As a result, the structure mapped into the generic space is a highly abstract pattern of spacial configuration of particular elements rather than any specific component inherent to both inputs.

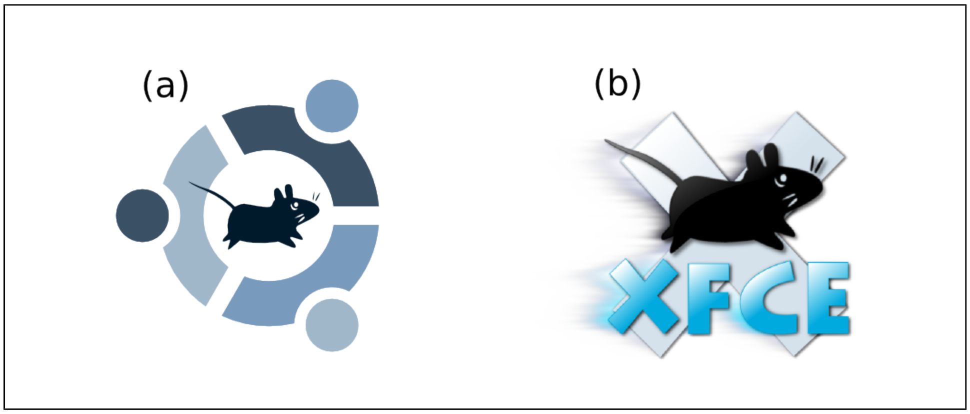

Figure 4. (a) Xbuntu logo; (b) XFCE logo.

[4] This is an instance of the already discussed mapping of additional elements from the input space, which are absent from the generic space and have no relevant correspondences across input spaces.

This schematic pattern is used for organizing the blend. The Ubuntu space contributes the circle of three and the XFCE space offers the image of a mouse; then, the mouse is placed in the same location as in the input space, i.e. in the center, which is conveniently left unoccupied in the Ubuntu logo. This particular location is forced by the optimality principle of topology, which stipulates that the configuration of elements in the blend should reflect the configuration in the input as closely as possible. Once again, the color of the novel sign is contributed by the desktop environment space and constitutes an “added” element, which does not belong to the generic space. The resulting blend is a coherent, self-contained visual image (the optimality principle of integration), whose elements can be easily traced back to appropriate inputs (the principle of unpacking).

Cases examined so far were rather unproblematic. Roughly speaking, the blend was a result of combining elements of two input signs according to a pattern derived from the generic space. This, however, does not apply to all Ubuntu derivative logos. Fig. 5(a) presents the logo of Fluxbuntu, a distribution with Fluxbox desktop environment,[5] whose logo is shown in Fig. 5(b). In this case, the content of the first input space is the Ubuntu logo, but the other space does not include any structures relating to Fluxbox. The content of this particular input space requires a brief explanation. The dominating graphic theme of Fluxbuntu (but not of Fluxbox!) is nature, which most probably is intended to become a distinguishing feature of this operating system. For instance, the dominant color in the graphic user interface is green, the default desktop background features leaves, etc. Consequently, the second input space contains graphic elements which are metonymically linked with the concept of nature, i.e. leaves and color green. Correspondences are drawn between the small circular elements at the edges of the circle of three and small roundish leaves. The content of the generic space is similar to the one of Kubuntu – both inputs contain a small, approximately round element.

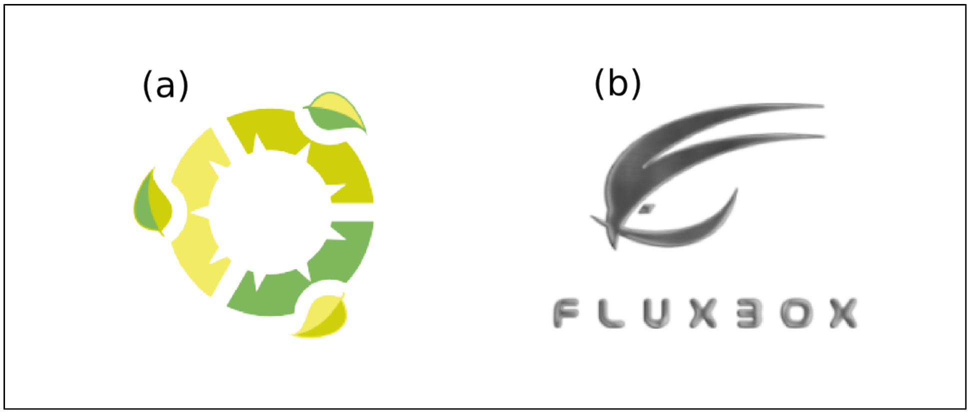

Figure 5. (a) Fluxbuntu logo; (b) Fluxbox logo.

[5] Technically, Fluxbox is not a fully developed desktop environment, but a window manager (cf. footnote 1). However, since technical details of software structure are not terribly important for the purpose of this article, for the sake of simplicity I will continue to use the imprecise term desktop environment.

In the blend the circular elements of the circle of three are elaborated into leaves in the very same way they were worked into gears in the Kubuntu blend. An additional element reinforcing associations with Fluxbuntu is the color recruited from the nature input space. Interestingly, the process of conceptual blending behind Fluxbuntu logo illustrates a different strategy of characterizing a Ubuntu-based distribution. So far, the distinguishing feature of various derivatives was the desktop environment included in the distribution. This is a convenient solution, as desktop environments differ in terms of appearance and functionality providing a convenient basis characterization of particular distributions. In principle, this path was open to Fluxbuntu as well due to the fact that this is the only Ubuntu-based distribution using this Fluxbox desktop environment. However, it is not a must for a logo to encode information about the desktop environment, especially when other salient features are available. In the case of Fluxbuntu, this feature is a unique graphic theme, which provides suitable input to build a recognizable logo.

The last non-linguistic sign discussed in this paper accompanies the distribution called Gobuntu. Its logo is presented in Fig. 6(a). As in all other cases, the "circle of three" from the Ubuntu logo is immediately recognizable. The other component is the well-known sign denoting the notion of recycling, provided in Fig. 6(b). The choice of this particular sign deserves a brief explanation. As already mentioned, almost all Linux-based operating systems are released under open licenses. Such software is free of charge and may be legally modified and reused by anyone. Most of Ubuntu derivatives, although released under open licenses, include components which may not be modified and therefore, strictly speaking, are not entirely open. Gobuntu, on the other hand, does not contain this type of components – all pieces of software are fully modifiable. According to the creators of the distribution, this "openness" is highlighted in the logo, as "the three arrows have a 'recycle' theme which describes the sharing and reuse of genuinely free software." (https://wiki.ubuntu.com/gobuntu_logo)

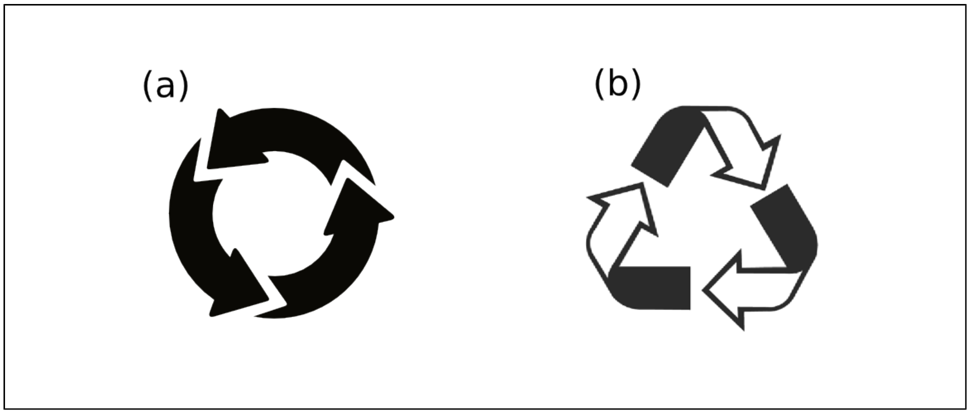

Figure 6. (a) Gobuntu logo; (b) recycling sign.

Let us now have a closer look at the content of mental spaces involved in the process of blending. Two inputs contain the Ubuntu logo and the recycling sign. Abstraction of the generic space structure is not problematic – it is a closed regular shape consisting of three equal, clearly distinguished sections. Exact shape of the figure is not specified in the generic space, but in the blend it is recruited from the Ubuntu space, while other details (arrows and lack of small circles at the edges) are contributed by the recycling space. Interestingly, in this case the color is not derived from any input and does not seem to play any significant role in the blend. Despite the fact that in Fig. 6(b) the sign is of the same color as Gobuntu logo, this similarity is incidental, because the recycling sign is not necessarily black. In fact, instances of this sign can have any color, green being perhaps most popular due associations with nature. The fact that color does not come from any input space may appear to be meaningless and, consequently, may be seen as violation of the optimality principle of good reason, but in fact the color is motivated indirectly. In previous logos the color of the blend highlights not only relationship with the desktop environment of graphic theme, but also the difference between the derivative and the original Ubuntu. Thus, Kubuntu and Xubuntu logos are blue not only because blue is the dominant color of the desktop environment space, but also because it is *not* the color of Ubuntu logo. In order to perform this function, the color had to be derived from the space other than the Ubuntu space. As the recycling sign does not have any fixed color, no specific color is available from the recycling space. On the other hand, the color still distinguishes Fluxbuntu from Ubuntu simply because it is *different* from the color of Ubuntu logo. In this sense, arbitrarily selected black performs the same function as motivated blue or green.

4 Blending and names

So far we have used conceptual blending to explain the formation of graphic signs. Logos, however, are only one element of "semiotic wrapping" of Ubuntu derivatives. In this section we will focus on linguistic signs used to denote particular distributions. It is perhaps not surprising that names of Ubuntu derivatives perform function similar to the one of logos – they underline common origin of all derivatives, but at the same time they point to salient differences between them. Perhaps even less surprising is the fact that the process governing the creation of novel names is once again conceptual blending.

Blending does not apply to the word *Ubuntu* [6] for the very same reasons it did not apply to Ubuntu logo – this distribution is the base for derivatives and its name does not have to encode any additional features distinguishing it from anything else. Instead, it is Ubuntu that other distributions need to be distinguished from. Once again we will begin with Kubuntu, whose name is probably the most straightforward case. *Kubuntu* is a combination of the *Ubuntu* and prefix *K*. A significant feature of almost all names of applications included in KDE desktop environment share certain morphological idiosyncrasy – they include letter *K*, e.g.:

(2) *Konqueror* (web browser)
(3) *KMail* (e-mail client)
(4) *KOffice* (office suite)
(5) *Kaffeine* (video player)

[6] In this section the words written in italics refer to the signifiers, i.e. spoken or written representations. Words enclosed with single quotation marks refer to meanings, i.e. semantic structures.

(6) *Amarok* (audio player)
(7) *KPDF* (PDF reader)
(8) *DigiKam* (video camera utility)

Whenever possible, the letter is integrated phonologically and orthographically into the word (sometimes against orthographic conventions of English), like in (2) and (5). Sometimes the letter appears in the middle, like in (8), the typical position, however, is the beginning of the word, like in (2), (3) and (4). Thus, letter *K* may be treated as "morpheme"[7] signaling relatedness of a particular piece of software with KDE.[8] However, the process of creating a novel name cannot be viewed as sheer affixation. Unlike typical affixes, the morpheme is quite unstable in terms of phonological and orthographic integration and distribution. Although *K* tends to be a prefix characterized by low degree of phonological and orthographic integration with the stem, like in (3), (4) and (7), it behaves rather opportunistically whenever closer integration is possible, like in (2), (5), (6) and (8). In other words, whenever the stem includes a grapheme *c* or *k* (equivalent to phoneme [k]), the *K* morpheme overrides the grapheme, usually at the expense of orthographic conventions of English. This suggests that the morpheme is used dynamically and its actual realization is not always predictable on the basis of rigid rules. Morphological peculiarities of this element are handled in a more satisfactory way by a dynamic process of conceptual blending.

Let us return to *Kubuntu* and accept a slightly oversimplified statement that one of the input spaces contains the word *Ubuntu*.[9] The other input includes morpheme *K* and specific words instantiating morphological patterns in which it occurs. We must realize that it is not sufficient to provide the morpheme alone – it should be accompanied with "instruction" determining use in specific contexts. For typical morphemes, this includes information about the distribution of the element (word initial, word final, word central, etc.). This, however, is not possible for morpheme *K*, as it is used more unpredictably and opportunistically. We must, therefore, accept that the "rules" are rather flexible and the actual realization is heavily dependent on additional factors.

The content of the generic space is a schematic structure of a word. This structure also determines the manner in which *K* combines with *Ubuntu*. Out of many morphological patterns instantiated in words in the KDE input (e.g. word initial *K* with no orthographic integration, word final *K* with integration, etc.) one pattern is selected, which is perceived as the most suitable for the stem *Ubuntu*. For instance, the pattern of well integrated word initial *K*, like in *Kaffeine*, is immediately rejected as in this word *K* is superimposed on an existing grapheme; for

[7] Technically speaking, K a real morpheme. The most important difference is the fact that real morphemes have fixed location within the word. In this case I use the term "morpheme" to refer to a meaningful particle that functions on the level of morphology.

[8] Notice that K is also a part of the KDE logo.

[9] At this point a question arises whether "word" stands for a phonological representation (actual cluster of sounds), written representation (actual group of graphemes) or a more abstract concept of a word (semantic structure). Although I believe that the last option is the best, answering this question falls outside the scope of this article and is not extremely relevant for our discussion. For the sake of simplicity, I will, somewhat arbitrarily, focus on written representations and ignore phonological nuances whenever possible.

Ubuntu, having no initial letter *k* or *c*, this option is not available. Eventually, the structure of the generic space is a pattern of a word containing no grapheme *k* or *c*, which applies both to *Ubuntu* and one of the words from the KDE space – the one found in, for instance, *KOffice*. In the KDE space words of this structure have the morpheme attached word initially. Thus, the content of the Ubuntu space facilitates the selection of appropriate morphological schema from the KDE input. The schema, in turn, is mapped into the blended space and governs the formation of the emergent structure – novel word involves morpheme *K* attached word initially to *Ubuntu*.

Kubuntu provided a useful scheme of name formation for future derivatives. Roughly speaking, the schema consists in attaching a prefix to *Ubuntu*, which functions as a morphological stem. Consequently, when Xubuntu was released, prefixation was already a preferred way of creating novel names. In terms of conceptual blending this pattern can be accounted for by means of proposing an auxiliary space embracing an abstract schema characterizing the morphological structure of *Kubuntu*. Thus, in the case of *Xubuntu* conceptual blending involves three input spaces: one of them contains word *Ubuntu,* another includes the already mentioned auxiliary schema and the last one is related with XFCE. Ideally, the XFCE space should contribute a prefix-like component, but this time the situation is slightly more complicated – there are no obvious candidates for this element. Certainly, the input contains a awkward cluster *XFCE*. However, unlike for KDE, there are rather few programs developed specifically for this desktop environment and the ones that are do not always mark relatedness with XFCE overtly. One example is Thunar, an XFCE file manager, whose belonging to this particular desktop environment cannot be deduced on the basis of the name. Other names do provide cues about XFCE, but they do it in various mutually incompatible ways. For example, a simple text editor for XFCE is Mousepad, its name alluding to the image of a mouse in the logo, but an application for writing data to DVD discs is Xfburn, which makes use of an arbitrarily chosen cluster *xf* from *XFCE*. However, *Xubuntu* does not choose any of these options; instead it selects the particle *x* used as a prefix. This strange choice can be accounted for by means of the contribution of the auxiliary space. In this blend, the auxiliary input provides not only a general morphological pattern of word formation (i.e. prefixation), but it also influences the internal structure of the prefix. To put it simply, the novel prefix is formed to resemble as closely as possible prefix *K*, i.e. a short, preferably one letter, element. In this way, morpheme *mouse*, used by *Mousepad*, is eliminated as too long. A better choice turns out to be a shorter cluster *xf*, but its internal structure is still too complex. Improved compatibility is achieved through clipping *xf* into *x*; hence *x* becomes the preferred prefix

Let us summarize the discussion on *Xubuntu*. The blending involves five spaces. Three of them are input spaces including Ubuntu input (with word *Ubuntu*), XFCE input (with word *XFCE*, *Mousepad*, *Xfburn*, *Thunar*, etc.) and an auxiliary input (with word *Kubuntu* determining certain pattern of word formation and internal structure of the prefix). There is a partial correspondence between morpheme *K* in auxiliary space and particle *xf* in XFCE space, which selects this particle from among other candidates. The generic space embraces a morphological pattern specifying little more than the notion of a component of a word being available for prefixation (this is probably the only property that can be safely claimed to be shared by all three inputs). The blended space contains a novel word *Xubuntu* composed of *Ubuntu* and *x-*, a prefix from XFCE input modified by the auxiliary pattern. The morphological process is guided by the pattern recruited from the auxiliary space. As we can see, the relations between particular spaces may become quite complex, especially when more than two input spaces are involved.

Fluxbuntu is much less complicated, although not less interesting. In the instances analyzed so far, both signs (the word and the logo) constituted a composition of Ubuntu component and a desktop environment component. However, the logo of Fluxbuntu is a combination of the circle of three and and graphic theme used in the derivative, while name follows the suite of *Kubuntu* and *Xubuntu*, selecting the desktop environment (i.e. Fluxbox) for one of the input spaces. We can only hypothesize about the reason for this discrepancy. The most plausible explanation seems to be the contribution of the third space containing *Kubuntu* (just like in the case of *Xubuntu*). By the time Fluxbuntu was released, the morphological schema of prefixation, started by *Kubuntu* and reinforced by *Xubuntu* had already become quite productive if not, to some extent, conventionalized.

In many respects the emergence of *Fluxbuntu*, follows the general pattern of the already discussed sign formation: there are three input spaces altogether, one of them is the Ubuntu space, another is Fluxbox space, the last is the auxiliary space with *Kubuntu*. In the second space *Fluxbox* is divided along the morpheme boundary into particles *flux* and *box*. The auxiliary space forces the prefixation for the preferred morphological process of word formation and the principle of topology selects *flux* rather than *box*, as the former appears word initially, in a prefix-like position in the input word. The generic space includes the notion of a morphological element available for prefixation. It is worth noticing that *Fluxbuntu* displays certain morphological peculiarity. The initial letter of Ubuntu is clipped, which in principle violates the pattern of prefixation provided by *Kubuntu* – the stem *Ubuntu* typically remains unaltered. Explanation of this peculiarity is provided by the internal morphological structure of *Fluxbox*, where the particle *flux* is immediately followed by *b*. As *b* is also one of initial letters of *Ubuntu*, the sequence *fluxb* from *Fluxbox* can be easily integrated into the blend at the expense of the initial letter. This phenomenon is not incidental (otherwise it would violate the principle of good reason), as, effectively, it amounts to strengthening the relation with *Fluxbox* – the import from this space is not only the morpheme *flux*, but also the manner of integrating it with the stem through *xb* cluster .

The last distribution covered in this article is Gobuntu. The motivation behind this name is far more hazy than in the case of previously analyzed Ubuntu derivatives. Similarly to Gobuntu logo, the name does not evoke any inherent part of the software (like the desktop environment); instead it contains reference to "philosophy" of developing and using open source applications. An important fact is that in *Gobuntu* particle *go* is sometimes written in green and *buntu* in black; this choice of colors is explained on one of Ubuntu websites (https://wiki.ubuntu.com/gobuntu_logo derived), where the authors write that "green in 'go' has an 'environmentally friendly' aspect, as well as reminding one of the green light in a traffic light, the common phrase that 'green means go'." In other words, *go* is to be associated with the permission to reuse and modify all components of the truly open operating system. Here, the network of mapping is somewhat more sophisticated, as *go* is associated with the notion of 'permissiveness' through the color of the letters in the enhanced graphical rendering of the name.

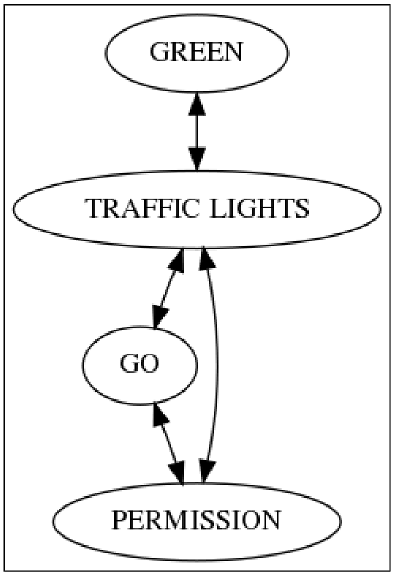

Figure 7.

Hence, apart from the Ubuntu input space the process of blending involves a broad space of permissiveness. This space includes a set various notions ('go', 'green', 'green means go', 'traffic light', 'permission', etc.). In some cases, the notion are semantically distant (e.g. 'go' and 'green'), but they are related via a system of metonymic connections sketched in Fig. 7. As evident from the diagram, green *go* in *Gobuntu*, is to trigger a chain of associations 'green' → '(green) traffic light' → 'go' → 'permission'.[10] In this way both green and *go* are metonymically linked with the notion of 'permission'. This semantic connection is exploited in the blended space – *go* is mapped into the blended space as an element metonymically associated with permissiveness (in graphically rich inscriptions, green strengthens this associations). The morphological pattern of prefixation is once again provided by auxiliary Kubuntu space. Similarly to Fluxbuntu, initial *u* of *Ubuntu* is dropped in the blend, but this time the reason is purely phonological – combination *ou* on the morpheme boundary leads to vowel hiatus and is rather awkward in English. In this case *u*-deletion should be treated as a mere phonological simplification rather than a meaningful contribution to the emergent structure.

This blend, however, is somewhat less optimal than the ones analyzed so far. One reason for this is the fact that it violates the optimality principle of unpacking

[10] In fact, both the network of the input space as shown in Fig. 7 and the chain of association presented in the text is more complex. In the space, the mappings are not perfectly symmetric, as suggested by two directional arrows, and in the chain the associations they are not perfectly unidirectional (for more extensive discussion on metonymy see Radden & Kövecses 1999). Moreover, as evident from the diagram, TRAFFIC LIGHTS is linked to PERMISSION not only through GO, but there is a direct metonymic connection between them.

– dense metonymic network in the permissiveness input space makes it virtually impossible to reconstruct connections between particular elements on the basis of the blend alone. As a result, the motivation behind particular components is quite obscure and the blend is not entirely transparent. Nevertheless, this does not make *Gobuntu* inadequate in any way; it simply means that the link between the signifier and the signified depends more heavily on convention rather than easily recognizable contributions from input spaces.

5 Restrictions on creativity

Conceptual blending, being a highly creative and dynamic process, opens many potential ways for combining seemingly dissimilar elements. So far we have focused on how various structures can be dynamically blended into one coherent sign. The questions that arises at this point is whether this creativity is unlimited, allowing for arbitrary composition of arbitrary elements, or is it restricted by some rules which limit the number of possible blends. If the latter is the case, what are these rules?

The notion of a rule requires a brief discussion. Rules are often thought of as strict, deterministic and inflexible laws, stipulating what *must* be done or what *must not* be done (or both). Within the field of linguistics, such understanding of rules is typical of the transformational-generative paradigm started by Noam Chomsky and virtually all structuralist schools. In cognitive linguistics, on the other hand, rules are conceived as **constructional schemas**, which embody articulatory routines and habits of speakers rather than abstract laws. Schemas, are less deterministic and prone to influences of other factors. Moreover, constructional schemas usually form a network and vividly interact with each other in the process of articulation. As a result, selection of particular schema is not a matter of applying fixed rules, but a result of a dynamic and not entirely predictable process. Schemas specify what *should* and what *is typically* done in order to achieve desirable communicative purpose, but violation of the schema does not always result in breakdown of communication.

In conceptual blending the source of restrictions are optimality principles, which function similarly to the constructional schemas. As we remember, optimality principles are not strict laws governing the process, but general "guidelines" which should be followed if the blend is to be understandable and cognitively economical. Most probably no single blend fully obeys all of principles; nevertheless, it can be safely assumed that more optimal blends (i.e. the blends that satisfy more optimality principles to greater degree) are preferred over less optimal ones. In the case of Ubuntu logos it is useful to propose two additional principles, which I will call **the optimality principle of balance** and **the principle of type**. According to the former, the contribution from the two main input spaces (typically the Ubuntu space and the desktop environment space) should be as equal as possible, so that the elements from one input space do not dominate the elements from the other. This principle is motivated by the chief function performed by a novel Ubuntu logo and names – on the one hand, the sign is to express, more or less equally, the fact of relatedness of a distribution to the Ubuntu family and point to a feature distinguishing it from the other members of this family. If this function is to be fulfilled successfully, elements from one input cannot become excessively salient, due to the risk of backgrounding the contribution from the other space. According to the principle of type, whenever possible, the blend should not mix structures of different types, i.e. non-linguistic graphic structures in logos should not be combined with linguistic elements and linguistic elements in names should not involve reference to non-linguistic elements.

Let us now have a look at how these principles work in practice. Kubuntu logo (Fig. 3(a)) combines the circle of three from the Ubuntu logo with the motif of a gear – the shape of the gear is superimposed on three small circle on the perimeter of the circle of three. This configuration reflects the principle of topology, since the overall shape of the small circles is similar to the overall shape of the gear. In principle, it is also possible to superimpose the gear motif on the whole circle of three, so that the logo features one big gear; in this case the principle of topology would be fulfilled equally well. Such a blend, however, would be much less optimal, as it would violate the principle of balance – there is a danger that an image of a big gear would produce very strong association with KDE and consequently dominate the Ubuntu component.

A similar situation is possible in the case of Xubuntu logo (Fig. 4). This blend involves the circle of three and the image of a mouse inside this circle. Another possible version of this sign would have the three images of a mouse replacing small circle on the perimeter and the inside of the circle empty. This, nevertheless, is also less optimal than the actual logo. The original sign follows the principle of topology, as here the mouse is placed in the same spatial configuration as in the XFCE logo; the alternative version would violate this principle and would not improve compliance with any other principle. Another version of Xubuntu logo could involve the combination of letters XFCE placed inside the circle of three instead of the mouse. This, however, would be at odds with the principles of type and topology (letters in XFCE logo are not in the center of the sign, but they would be in such position in the alternative Xubuntu logo). This principle also explains why the name of Xubuntu is not *Mousebuntu*; potentially this path was open for the authors of this distribution, since the image of a mouse is very characteristic of XFCE (definitely more characteristic than morpheme *X*, which in the world of Linux operating systems tends to be associated with other types of software). *Mouse-* in *Mousebuntu*, would refer to non-linguistic graphic elements of the desktop environment, which, according to the principle of type, should not be expressed in linguistic signs.

The optimality principle of balance accounts for the fact that in all examples the color of the blended sign is not inherited from the Ubuntu input space. Since in all examples this space contributes a very salient motif of the circle of three, the color is derived from the other space in order to strengthen the connection with this space. For the already discussed reasons, the principle is operative even in the case of the seemingly unmotivated color of Gobuntu logo. Hence, the outcome of conceptual blending is the result of tension between dynamic creativity and the restrictions imposed by the principles of optimality. The choice of a particular blend out of many potentially possible structures is not random; instead, the most optimal blends (as specified by the principles of optimality) are favored over less optimal ones.

6 Beyond Ubuntu logos

Logos of Ubuntu derivatives are a perfect illustration of how conceptual blending can systematically organize the process of sign formation within a group of related signs. Despite the fact that there are no fixed rules determining precisely how two signs should be blended and in each case the details of the process are dependent on characteristics of the input signs, combining components of the default Ubuntu logo with other graphic structures has become a standard manner of creating logos of derivative distributions. However, conceptual blending is not restricted to the sign formation within this narrow group. In fact, this cognitive process accounts for a number various visual signs.

One of the best known examples is the Union Jack – the flag of the United Kingdom (Fig. 8). This sign is a combination of three other signs: the flags of England, Scotland and Ireland (with an additional “transitional” flag from the seventeenth century). In this case the motif of a cross present shared by all flags is mapped without fusion into the blend and contribution of colors mapped faithfully from the inputs reinforces the resemblance to each of the flags. The sign is to symbolize the close union of England, Scotland and Ireland, in which the three countries retain much of their national characteristics.

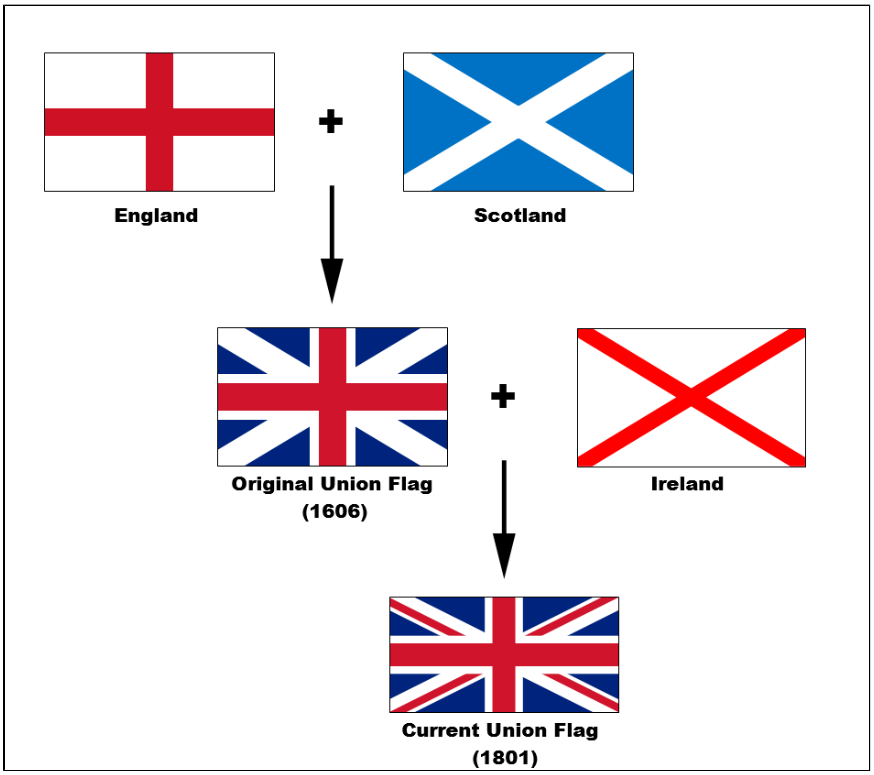

Figure 8.

Another “vexillological” example of the flag of National Bolsheviks, a extremist political movement that claims to combine the ideology of radical nationalism and Bolshevism. Both elements are expressed semiotically in the national Bolshevik flag presented in Fig. 9(a), an obvious blend of the Nazi flag (Fig. (b)) and the motif of the hammer and sickle associated with the communist movement (Fig. (c)). In this case, the distinctive element of the hammer and sickle is mapped into the region originally occupied by the swastika, but the color is derived from the Nazi flag space to strengthen the semantic link with the radical nationalist emblem.

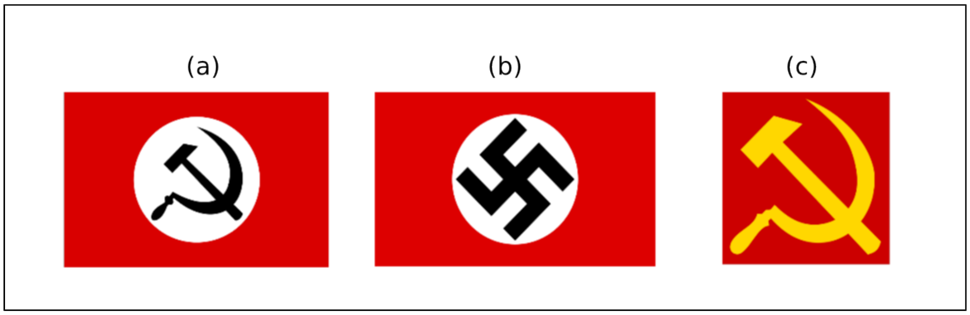

Figure 9.

Blends of graphic signs are sometimes used to highlight alleged or fictional connections between institutions or to criticize a policy of a company. Figure 10(a) demonstrates a fictional logo of what could be a space agency established or sponsored by IT corporations. This sign blends two well known logos – one of them is the National Aeronautics and Space Administration (NASA; Fig. (b)), while the other is one of the biggest IT companies, Apple Inc. (Fig. (c)). Here, only the overall shape of an apple is mapped from the Apple Inc. input space, whereas other details are provided by the NASA input, however, the apple shape is so distinctive that it constitutes a sufficient connection with the Apple Inc. space.

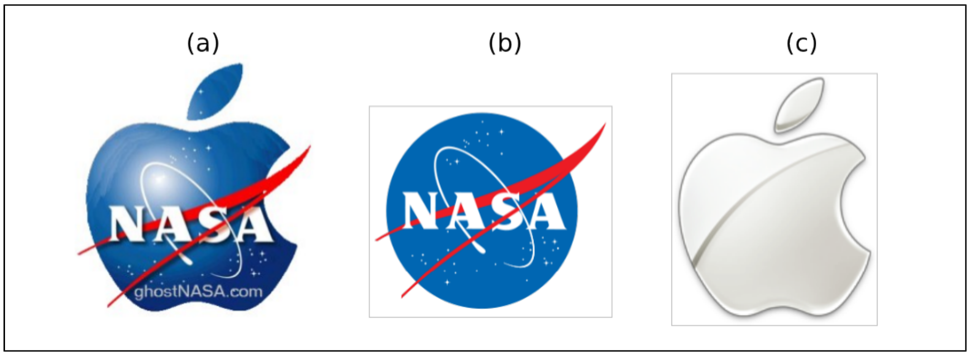

Figure 10.

Figure 11(a) in turn features a humorous blend of the Eastern States Standard Oil (also known as Esso), whose logo is presented in Fig. (b). This blend exploits the similarity between letter S and the dollar symbol, as a result of which the symbol is mapped into the Esso logo. This combination seems to suggest the excessive preoccupation with money on the part of the company.

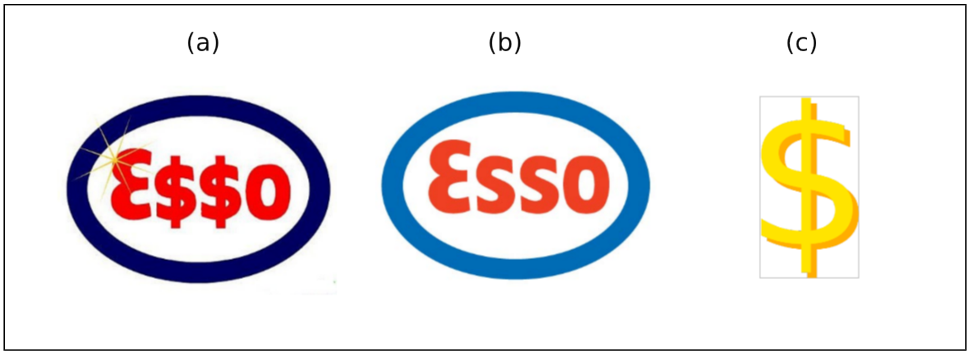

Figure 11.

Blends are of logos are often used for purely humorous purposes. The final example brings us back to the field of information technology. The sign is a logo of a fictional operating system Macintosh Windows (Fig. 12(a)), a hybrid of Microsoft Windows (its logo is shown in the Fig. 12(b)) and Macintosh's Mac X OS operating systems (produced by Apple Inc.; its logo is presented in Fig. 10(c)). It is worth mentioning that in this case, the name of the fictional system is a result of blending as well. The comic effect is evident if we bear in mind that Microsoft and Macintosh are the main rivals in the market of computer operating systems and it is virtually impossible for them to cooperate in order to release a such a hybrid piece of software.

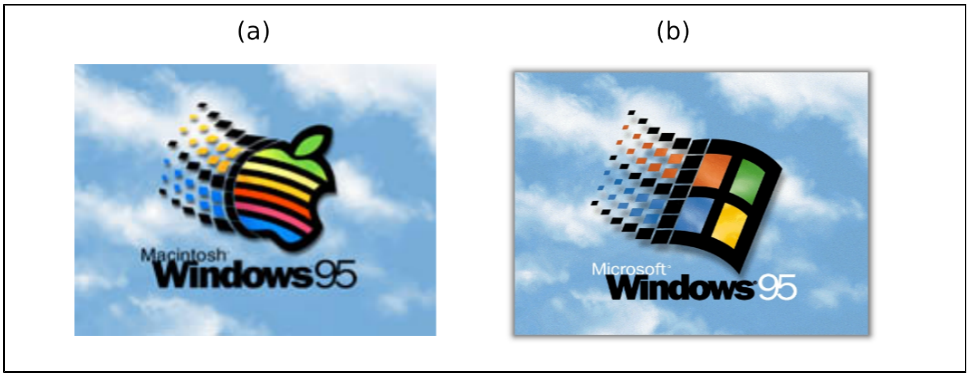

Figure 12.

In spite of the fact that the blending theory is most widely used in the field of linguistics, the process is cognitive rather than linguistic in nature. The above example prove that elements of graphic signs may be successfully blended and used for a variety of purposes. More importantly, the exemplar of Ubuntu derivatives shows that in some cases conceptual blending can be used persistently and systematically for certain array of signs.

7 Conclusion

Logos and names of many Ubuntu derivatives are formed in roughly the same way. Although the kind of input used for each type of sign is different (graphic structures vs. morphemes), the basic mechanisms responsible for blending of the material are very similar. In both cases the sign is a composition of elements from Ubuntu space (morpheme *Ubuntu* vs. the circle of three) and element related with distinctive features of each distribution (morphemes vs. graphic structures associated with desktop environment, graphic theme, etc.). Regardless of the type of the material, similarities are sought across available structures; these similarities are used later as a basis for constructional operations of blending existing elements into novel semiotic structures. New signs are tightly knit and coherent units, which inherit recognizable components from original signs and are usually semantically transparent.

The analysis proposed in this article is by no means exhaustive. Intimately related with the points discussed above is the question of motivation, i.e. the non-arbitrariness of signifier-signified link. This matter was only touched upon in several places, where it was relevant for the discussion, but it definitely deserves a more complete treatment. Another issue that did not receive sufficient attention is the question of medium (speech vs. writing) and its impact on word formation (one example of such influence is *u*-deletion in *Gobuntu*). Also the distributions selected for the analysis are just several instances of all existing Ubuntu derivatives. However, in my opinion, the examples are adequate illustrations of various phenomena in the process of conceptual blending and convincingly demonstrate similarities in the formation of linguistic and non-linguistic signs.

As we have seen, creation of new signs is a dynamic process. Conceptual blending cannot *predict* its ultimate outcome, but it is able to *describe* it in a fairly detailed manner. This should not be considered as a shortcoming of the theoretical framework, as probably there are no fixed, deterministic and universal rules governing formation of novel sign (especially non-linguistic ones, like logos). In many cases the process is highly creative and utilizes opportunistically many types of seemingly unrelated structures (e.g. the circle of three and an image of a mouse in Xubuntu logo). The theory of conceptual blending provides a convenient descriptive model, which allows us to define general principles structuring this dynamic and seemingly chaotic conceptual operations (optimality principles). Another strength of conceptual blending is that it is able to systematically account for apparently incidental or arbitrary phenomena (e.g. *u*-deletion in *Fluxbuntu*).

At the same time, this powerful descriptive tool does not employ highly sophisticated and abstract apparatus. In spite of the fact that actual realizations of the process may be quite complex, conceptual blending is founded on fairly basic cognitive capabilities of human mind – ability to perceive similarities and extract patterns embodying them, establishing correspondences between entities, combining conceptual structures, etc. (cf. e.g., Langacker 1982). These capabilities are not restricted to the processing of one type of data; consequently, conceptual blending may be applied successfully to the analysis of many types of data.

References

Fauconnier, Gilles, 1985. Mental Spaces. Cambridge: MIT Press.

Fauconnier, Gilles, & Turner, Mark, 2007. Conceptual integration networks. The Cognitive Linguistics Reader (2007). Evans, Vyvyan, Bergen, Benjamin K, & Zinken, Jörg (eds.). London-Oakland: Equinox.

Grady, Joseph E., Oakley, Todd, Coulson, Seanna, 2007. Blending and metaphor. The Cognitive Linguistics Reader (2007). Evans, Vyvyan, Bergen, Benjamin K, & Zinken, Jörg (eds.). London-Oakland: Equinox.

Langacker, Ronald W., 1982. Space grammar, analysability, and the English passive. Language 58 (1982) 22-80.

Oakley, Todd, 1998. Conceptual blending, narrative discourse, and rhetoric. Cognitive Linguistics 9 (1998) 321-360.

Radden, Gunter & Kövecses, Zoltan, 1999. Towards a Theory of metonymy. Metonymy in language and Thought (1999) Panther, Klaus-Uwe & Radden, Gunter (eds.) Amsterdam: John Benjamins.

Turner, Mark & Fauconnier, Gilles, 1995. Conceptual integration and formal expression. Metaphor and Symbolic Activity 10 (1995) 183-204.

Turner, Mark & Fauconnier, Gilles, 2003. Metaphor, metonymy, and binding. in Barcelona, A. (ed.) Metaphor and Metonymy at the Crossroads (2003) New York: Mounton de Gruyter.

Sources of images

- Ubuntu and Kubuntu logos derived from https://wiki.ubuntu.com/Artwork/Official
- GNOME logo derived from http://gnome.org
- KDE logo derived from http://kde.org
- Xubuntu logo derived from http://lacasadetux.files.wordpress.com/2008/05/xubuntu_logo.png?w=200&h=203
- XFCE logo derived from http://www.xfce.org/about/artwork
- Fluxbuntu logo derived from http://ubuntu.onego.ru/wp-content/uploads/2008/02/fluxbuntu_logo-vi.png
- Fluxbox logo derived from http://www.linuxguiden.no/images/5/52/Fluxbox_logo.png
- Gobuntu logo derived from http://commons.wikimedia.org/wikipedia/commons/d/db/Gobuntu_logo_no_text.png
- Recycling sign derived from http://www.fs.utoronto.ca/Assets/recycle.gif
- Union Jack image logo derived from http://olivetteotele.files.wordpress.com/2006/12/flags_of_the_union_jack.png
- National Bolsheviks flag, swastika flag, NASA logo and Apple Inc. logo derived from http://en.wikipedia.org
- NASA-Apple logo derived from http://www.ghostnasa.com/posts/027applenasa.html
- Parody of Esso logo derived from http://www.greenpeace.org/raw/image_full/international/photosvideos/photos/parody-of-esso-logo
- Esso logo derived from http://www.petchem.co.th/images/Customer%20group%20&%20logo/Refinary/Esso%20LOGO.jpg
- Dollar symbol derived from http://grantblackley.net/DollarSymbol2.png
- Macintosh Windows logo derived from MacWindows from http://www.kevcom.com/images/macwindows/macwindows.jpg
- Microsoft Windows logo derived from http://www.psp-themes.net/data/media/10/Windows%2095.jpg