

Винник О. Ю.,

Львівський національний університет імені Івана Франка, м. Львів

ОБРАЗНІСТЬ У СУЧАСНОМУ АНГЛОМОВНОМУ ДИСКУРСІ ПРОГРАМУВАННЯ

У статті запропоновано аналіз функціонування образних елементів у сучасному англomовному дискурсі програмування. Виявлено, що образні одиниці, зокрема, метафори та порівняння, виконують у цьому типі дискурсу функцію пояснення та популяризації професійного знання.

Ключові слова: англomовний дискурс програмування, образність, метафора, порівняння.

В статье предложен анализ функционирования образных элементов в современном англоязычном дискурсе программирования. Обнаружено, что образные единицы, в частности, метафоры и сравнения, выполняют в этом типе дискурса функцию разъяснения и популяризации профессионального знания.

Ключевые слова: англоязычный дискурс программирования, образность, метафора, сравнение.

The paper proposes analysis of functioning of figurative expressions in modern English programming discourse. The study has revealed that the function of figurative language, in particular, metaphor and simile, in this type of discourse consists in explanation and popularization of professional knowledge.

Key words: English programming discourse, imagery, metaphor, simile.

Однією з найперспективніших і впливових у сучасному вітчизняному мовознавстві є когнітивно-комунікативна парадигма, центральною одиницею вивчення якої є *дискурс*. Цим і пояснюється незмінний інтерес вітчизняних лінгвістів до проблеми аналізу різних типів дискурсів, зокрема, тих, що стосуються професійної діяльності людини (див., напр., [Ананко 2007, Біскуб 2009, Васік 2008, Вострова 2003, Колеснікова 2009, Шукало 2008, Юшковець 2008]).

Об'єктом вивчення у цій статті є сучасний англomовний дискурс програмування, який на сьогодні усе ще залишається маловивченим типом мовленнєвої комунікації. Дослідження спрямовано на аналіз функціонування образних елементів у цьому виді професійного дискурсу. Фактичним матеріалом послуговували 8 англomовних навчальних книг з програмування за 1993 – 2009 роки.

Основою традиційного підходу є думка про те, що для професійного дискурсу специфічною властивістю є структурованість, логічна аргументованість й об'єктивність викладу спеціальної інформації. Тому нестандартність її презентації, поширення емоційно забарвлених елементів тощо є явищами контрастними, небажаними [Колеснікова 2009]. Тим часом навіть поверхневий аналіз текстів англomовного дискурсу програмування дозволяє виявити в них велику кількість іншостильових та індивідуально-авторських елементів, зокрема, тропів. Основою образності в аналізованих текстах переважно виступає аналогія, тобто схожість предметів чи явищ за якимись ознаками. Одним із загальновідомих її мовних втілень є *метафора*. У текстах англomовного дискурсу програмування метафора виконує функцію роз'яснення складних професійних понять на базі їх схожості з предметами й явищами нашого повсякденного життя. У наведеному нижче прикладі

Iterators are the glue that holds containers and algorithms together. [Stroustrup 1997, p. 549].

ітератори ототожнюються з клеєм на основі схожості виконуваних ними функцій. Метафора виконує у цьому реченні характеризуючу функцію та займає предикатну позицію.

За структурною організацією метафори в англomовному дискурсі програмування поділяються на *прості*, у яких план вираження представлений однією одиницею, та *розгорнуті* (метафоричні ланцюжки), які об'єднують у своєму складі групу тематично зв'язаних простих метафор. Наведемо приклад, який добре ілюструє поняття розгорнутої метафори:

Yes, I think object-oriented programming is a religion. I've seen programming religions come and go: structured programming, modular programming, top-down programming, Each religion has its obvious merits; none is complete, permanent or perfect. Some people are more easily saved than others; I've seen enough of these religions rise and fall to have developed some cynicism about them, despite their attractions. It's the too easily saved souls with messianic gleam in their eyes and an utter lack of practical experience, who scare me. [Heller 1993, p. 70].

Це чотиричленна розгорнута метафора, представлена словами «релігія» (*religion*), «рятувати» (*to save*), «душа» (*soul*) та «месіанський» (*messianic*). Центральним образом у ній є релігія, референтом якої є деякий стиль програмування (об'єктно-орієнтоване програмування, структурне, модульне тощо), все решта становить метафоричну розгортку.

Відповідно до тематичної співвіднесеності порівняння, що лежить у його основі, образні метафори в англomовному дискурсі програмування можна умовно поділити на більш чи менш значні за об'ємом тематичні групи, наприклад, метафори антропоморфні, «архітектурно-будівельні», «географічні», «кулінарні», «автомобільні», «філософсько-релігійні» тощо.

Антропоморфні метафори – це метафори, в основі яких лежить уподібнення людині, наділення людськими психічними якостями елементів програмного забезпечення та процесу його розробки. Як засвідчує аналіз мовного матеріалу, практично будь-який компонент процесу програмування, реальний чи абстрактний, може поводити себе як людина: розмовляти (*If certain areas are at risk of failing to meet their goals, the architecture should say so.* [McConnell 2004, U3, p. 26]), щось (не)знати (*The compiler knows where to find this and other system files* [Pohl 2002, p. 4], *Because the compiler does not know the meaning of a userdefined +,...* [Stroustrup 1997, p. 265]), нервуватися (*the C compiler is very fussy about casting.* [Heller 1993, p. 9]), нагадувати (*The compiler diagnostics reminded me that*

I had to copy the new function prototypes to IMAGE3. H; ... [Heller 1993, p. 30]), самостійно вирішувати й мати власні бажання (*Objects can even decide at runtime whether they want to be kept informed.* [Freeman 2004, p. xii]) тощо.

Антропоморфність, тобто надавання неживим предметам властивостей мислячих людських істот, набула значного поширення у сучасних науково-технічних, зокрема, комп'ютерних науках. Ставлення до комп'ютера як до живої істоти давно стало звичним для тих, хто постійно користується ним у повсякденній побутовій та професійній практиці. У сучасному дискурсі програмістів ми спостерігаємо подальше розгортання концептуальної метафоричної моделі «Комп'ютер – це людина» у напрямку наділення людськими емоціями, думками, мовою елементів, які мають відношення до програмного забезпечення комп'ютера та його розробки.

«Архітектурно-будівельні» метафори – це метафори, джерелом яких є сфера архітектури та будівництва:

If people had infinite time, resources and space to build physical structures, you would see incredible sprawling buildings with one room for each shoe and hundreds of rooms. This is how software is developed. [McConnell 2004, U 5, p. 4]

«Географічні» метафори – це метафори, сферою походження яких є географія. Наведений нижче приклад ілюструє типовий випадок вживання цих метафор, а саме, представлення навчального процесу з програмування чи самого процесу розробки програмного забезпечення як нелегкої подорожі по незнайомій місцевості у супроводі провідника, що ним є сам автор:

It's been quite a trip. Here we are in the promised land of 32-bit programming, revealing in Windows NT and Win32s. It's a shame that we have to keep returning to the old world of 16-bit programming to visit our relatives who still live in plain Windows. [Heller 1993, p. 441]

«Кулінарні» метафори – це метафори, джерелом яких є галузь кулінарії. Наприклад, говорячи про розробку програмного забезпечення з інтерфейсами, що взаємодіють між собою, автор книги «Interface-Oriented Design» постійно звертається до тем, пов'язаних з піцою (за нашими підрахунками слово «піца» вжито у книзі 372 рази). У наступному прикладі автор використав аналогію з піцою для того, щоб проілюструвати властивості та дії, які можна застосувати до об'єкта класу:

If you were making a pizza yourself, rather than ordering one, you might have a class like this:

```
class SimplePizza
set_size(Size)
set_topping(Topping)
Size get_size()
Topping [] get_toppings()
make()
```

At the completion of the make() method, a SimplePizza is ready for your eating pleasure. [Pugh 2006, p. 62].

«Автомобільні» метафори – це метафори, джерелом яких є галузь автомобілебудування, наприклад:

One way to think about a car is as a collection of wheels, doors, seats, windows, and so forth. Another way is to think about what a car can do: It can move, speed up, slow down, stop, park, and so on. A class enables you to encapsulate, or bundle, these various parts and various functions into one collection, which is called an object. [Liberty 2006, Day 6].

У наведеному прикладі значення таких базових понять об'єктно-орієнтованого програмування, як «клас» (*class*) та «об'єкт» (*object*), пояснюється на прикладі різних деталей та дій машини.

«Філософсько-релігійні» метафори – це метафори, джерелом яких є філософія та релігія. Наприклад, назва першого розділу книги «A Project Guide to UX Design» звучить як *The Tao of UXD* (укр. *Дао проектування досвіду взаємодії*). *Дао* – одне з центральних понять китайської філософії; першооснова світу. У розділі автор роз'яснює суть базового поняття книги, а саме, проектування досвіду взаємодії, та його важливість.

Згідно з традиційним підходом, образна метафора є авторською, унікальною. Вона не переходить у словник, що підтверджує її динамічність і зв'язок з індивідуальною картиною світу, тобто з особистісним тезаурусом носія мови. Саме тому образну метафору не завжди можна легко розгадати або ж вони дають привід для різної їх інтерпретації [Телия 1988]. У художній літературі образна метафора виконує емоційно-естетичну, декоративну функцію. У науковому дискурсі метафори виступають передовсім у функції популяризації наукового знання та роз'яснення значення наукових понять, тому їх ще називають педагогічними, пояснювальними, дидактичними [Москвин 2000, с. 72]. При цьому метафора тісно сплітається з усією структурою наукового тексту і, як наслідок, допускає лише однозначне тлумачення, що відрізняє її від можливості широкого тлумачення в художньому творі [Дядюра 2001, с. 10]. Функції метафори у професійному дискурсі схожі на її функції у дискурсі науковому. Це однак не означає, що, потрапивши у спеціальний текст, метафора втрачає свою образність. Навпаки, образна метафора в професійному дискурсі – це ніби яскравий штрих на сірому тлі серйозних фактів і гіпотез; це своєрідна призма, яка перетворює наповнені незрозумілим змістом складні об'єкти, процеси, явища в усім відомі й звичні речі.

Образне порівняння – фігура мовлення, що полягає у зіставленні одного предмета або явища з іншим на основі деякої спільної для них ознаки. Як і образні метафори, порівняння в англомовному дискурсі програмування переважно виконують функцію пояснення, наприклад:

A class is a lot like an iceberg: Seven-eighths is under water, and you can see only the one-eighth that's above the surface. [McConnell 2004, U5, p. 22].

У цьому прикладі автор влучно порівнює клас з айсбергом. В основі порівняння лежить той факт, що більшість аспектів проектування чи конструювання класу є недоступними, «невидимими» для інших класів. Порівняння вербалізується у значенні апроксиматора-порівнювача *a lot like* і слугує для вираження відносної подібності між предметами.

Порівняння має багато спільного з метафорою. Однак між ними існує низка відмінностей. Наприклад, порівняння у своєму класичному варіанті тричленне (А схоже з В за ознакою С), тоді як метафора – двочленна (А є В).

Порівняння вказує на схожість двох об'єктів незалежно від того, чи ця схожість постійна чи тимчасова, дійсна чи уявна, обмежена одним аспектом чи глобальним, тоді як метафора виражає постійну ознаку предмета тощо.

Типова сфера функціонування образних засобів – розмовно-побутове та літературне мовлення. Цілком очевидно, що їх поява в англomовному дискурсі програмування пов'язана з намаганням автора полегшити читачеві складний інтелектуальний процес опанування знаннями в галузі програмування, «розвантажити» мозок адресата, переключивши його увагу на теми, що не мають безпосереднього відношення до програмування, а також вплинути на емоції читача, розвеселити, розважити його для того, щоб вивчення програмування принесло не лише користь, але й задоволення.

Проведене дослідження окреслює нові ракурси аналізу професійного дискурсу в плані функціонування у ньому позажанрових та іншостильових елементів. Подальші наукові розвідки можуть бути спрямовані на більш детальне висвітлення структурно-композиційної та смислової специфіки реалізації категорії образності в англomовному дискурсі програмування, а також в інших видах професійного дискурсу на матеріалі різних мов.

Література:

1. Ананко Т. Р. Англomовний корпоративний дискурс: автореф. дис. на здобуття наук. ступеня канд. філол. наук: спец. 10.02.04 «Германські мови» / Ананко Тетяна Рудольфівна; Харк. нац. ун-т ім. В. Н. Каразіна. – Х., 2007. – 20 с.
2. Біскуп І. П. Англomовний дискурс програмного забезпечення як модель мовленнєвої взаємодії людини й комп'ютера: монографія / І. П. Біскуп. – Луцьк: Волин. нац. ун-т ім. Л. Українки, 2009. – 388 с. – ISBN 978-966-600-473-7.
3. Васік Ю. А. Ритмічна організація англomовного політичного дискурсу (експериментально-фонетичне дослідження на матеріалі промов сучасних політиків Великої Британії): автореф. дис. на здобуття наук. ступеня канд. філол. наук: спец. 10.02.04 «Германські мови» / Васік Юлія Анатоліївна; Донец. нац. ун-т. – Донецьк, 2008. – 19 с.
4. Вострова С. В. Лінгвокогнітивні та комунікативно-прагматичні особливості сучасного англomовного медичного дискурсу (на матеріалі медичних текстів з проблематики ВІЛ/СНІДу): автореф. дис. на здобуття наук. ступеня канд. філол. наук: спец. 10.02.04 «Германські мови» / Вострова Світлана Володимирівна; Київ. нац. ун-т ім. Т. Шевченка. – К., 2003. – 20 с.
5. Дядюра Г. М. Функціональні параметри образності в науковому стилі (на матеріалі текстів природничих та технічних наук): автореф. дис. на здобуття наук. ступеня канд. філол. наук: спец. 10.02.01 «Українська мова» / Дядюра Галина Миколаївна; Ін-т укр. мови НАН України. – К., 2001. – 22 с.
6. Колеснікова І. А. Лінгвокогнітивні та комунікативно-прагматичні параметри професійного дискурсу: автореф. дис. на здобуття наук. ступеня д-ра філол. наук: спец. 10.02.15 «Загальне мовознавство» / Колеснікова Ірина Анатоліївна; Київ. нац. ун-т ім. Т. Шевченка, Ін-т філології. – К., 2009. – 32 с.
7. Москвин В. П. Русская метафора: параметры классификации / В. П. Москвин // Филологические науки. – 2000. – № 2. – С. 66-74.
8. Телия В. Н. Метафоризация и её роль в создании языковой картины мира // Роль человеческого фактора в языке. Язык и картина мира / отв. ред. Б. А. Серебренников. – М., 1988. – С. 173-203. – ISBN 5-02-010880-4.
9. Шукало І. М. Англomовний комп'ютерний рекламний дискурс: семантичний та комунікативний аспекти (на матеріалі веб-сайтів британських туристичних компаній): автореф. дис. на здобуття наук. ступеня канд. філол. наук: спец. 10.02.04 «Германські мови» / Шукало Ірина МIRONІВНА; Київ. нац. лінгв. ун-т. – К., 2008. – 20 с.
10. Юшковець І. А. Комунікативні стратегії і тактики в політичному дискурсі канцлерів ФРН (на матеріалі урядових заяв і політичних виступів): автореф. дис. на здобуття наук. ступеня канд. філол. наук: спец. 10.02.04 «Германські мови» / Юшковець Інна Анатоліївна; Донец. нац. ун-т. – Донецьк, 2008. – 20 с.

Список джерел ілюстративного матеріалу:

1. Freeman E., Robson E., Bates B., Sierra K. Head First Design Patterns / Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra. – O'Reilly Media, 2004. – 678 p. – ISBN 0-596-00712-4.
2. Heller M. Advanced WIN32 Programming / M. Heller. – John Wiley & Sons, Inc., 1993. – 463 p. – ISBN 0-471-59245-5.
3. Liberty J. Sams Teach Yourself C++ in 21 Days / J. Liberty. – 4th. ed. – Sams, 2006. – 912 p. – ISBN 0-7686-5751-2.
4. McConnell S. Code complete: a practical handbook of software construction / S. McConnell. – Redmond, Wash. : Microsoft Press, 2004. – 960 p. – ISBN 0-735-61967-0.
5. Pohl I. C++ by Dissection / I. Pohl. – Addison-Wesley, 2002. – 501 p. – ISBN 0-201-74396-5.
6. Pugh K. Interface Oriented Design / K. Pugh. – The Pragmatic Programmers LLC, 2006. – 240 p. – ISBN 0-9766940-5-0.
7. Stroustrup B. The C++ Programming Language / B. Stroustrup. – 3rd. ed. – AT&T Labs, 1997. – 912 p. – ISBN 0201889544.
8. Unger R., Chandler C. A Project Guide to UX Design: For User Experience Designers in the Field or in the Making / R. Unger, C. Chandler. – New Riders Press, 2009. – 288 p. – ISBN 0-321-60737-6.