УДК 378:81’255.2:6:[811.161.2+811.111]

Л.М. ЧЕРНОВАТИЙ,
професор кафедри теорії та практики перекладу англійської мови Харківського національного університету імені В.Н. Каразіна

Ю.В. КУПРІЄНКО,
викладач кафедри ділової іноземної мови та перекладу національного технічного університету «Харківський політехнічний інститут»

ПОНЯТТЄВА СХЕМА ЯК ОСНОВА ФОРМУВАННЯ ПРЕДМЕТНОЇ ТА ТЕРМІНОЛОГІЧНОЇ СКЛАДОВИХ ФАХОВОЇ КОМПЕТЕНТНОСТІ ПЕРЕКЛАДАЧІВ ГАЛУЗЕВИХ ТЕКСТІВ

Наведено опис процесу складання поняттєвої схеми та відбору англо- й україномовної термінології (галузь – «Комп’ютерні науки», тема – «Структури даних») для формування предметного й термінологічного компонентів фахової компетентності перекладача як етапу розробки експериментальної методики навчання галузевого перекладу.

Ключові слова: навчання галузевого перекладу, комп’ютерні науки, структури даних, фахова компетентність перекладача, предметна компетентність, термінологічна компетентність, англійська й українська мови, поняттєва схема.

Постановка проблеми та аналіз останніх досліджень. Фахова компетентність перекладача (ФКП), що у найбільш загальному вигляді тлумачиться як сума знань, навичок і умінь, необхідних для здійснення перекладу на фаховому рівні (див. також [2]), є метою навчання майбутніх перекладачів. ФКП включає декілька компонентів, конкретний перелік яких є дискусійним (див. наприклад, [1; 3; 5]). У межах даного дослідження ми спиратимемося на робочу модель ФКП, запропоновану в інших наших працях [2]. Згадана модель включає п’ять компетентностей: білінгвальну, екстралінгвістичну, перекладацьку, особистісну та стратегічну, з яких у цій розвідці приділимо більшу увагу екстралінгвістичній та білінгвальній. Екстралінгвістична компетентність включає всі фонові (енциклопедичні, тематичні, соціобікультурні тощо) і предметні (інформація щодо поняттєвого складу певної галузі людської діяльності та міжпоняттєвих зв’язках) знання, з яких зосередимося на предметних, оскільки наше дослідження стосується галузевого перекладу. Як зазначалося у наших попередніх роботах [2], предметна складова загалом тлумачиться як певний діапазон знань, необхідний для перекладу у відповідній сфері, без наявності якого переклад суттєво ускладнюється або взагалі унеможливлюється. Згадані знання утворюються сумою понять, на яких ґрунтується відповідна галузь, а кожне поняття в кожній мові позначається відповідними термінами, володіння якими є необхідною передумовою галузевого перекладу. Термінологічний компонент входить до складу білінгвальної (зокрема, мовної) компетентності. Таким чином, з одного боку, якість галузевого перекладу суттєвим чином залежить від якості сформованості предметного та термінологічного субкомпонентів ФКП. З іншого боку, системні дослідження їх формування в межах компетентнісного підходу на сьогодні фактично відсутні, що й зумовлює *актуальність* нашого дослідження.

Формулювання мети. *Метою* статті є опис досвіду складання (на основі англомовного тексту) поняттєвої схеми для засвоєння головних понять, на яких ґрунтується певний

сегмент галузі (комп'ютерна сфера), і виділення корпусу україно- та англомовних термінів, засвоєння яких є необхідним для перекладу відповідних англомовних текстів. Згадані складання й виділення є кроками на шляху розробки експериментального комплексу вправ для навчання професійно орієнтованого перекладу у згаданій сфері. Відповідно *об'єктом* дослідження є поняттєвий аналіз галузевих текстів з метою відбору головних понять та встановлення їх ієрархії, а *предметом* – розробка поняттєвої схеми згаданого сегмента (структура даних) та відбір двомовної термінології для позначення відповідних понять.

Виклад основного матеріалу. Для досягнення згаданої мети необхідно було розв'язати низку *завдань*: 1) визначити сферу та сегмент дослідження; 2) підібрати репрезентативний англомовний текст оригіналу (ТО) для перекладу та аналізу; 3) створити текст перекладу (ТП) українською мовою; 4) виділити в обох текстах терміни-відповідники; 5) скласти англо-український та українсько-англійський термінологічні словники; 6) відібрати головні поняття, на яких ґрунтуються ТО і ТП, і встановити їх ієрархію; 7) виходячи з цього, скласти поняттєву схему ТО й ТП і проаналізувати її; 8) окреслити перспективи подальшого дослідження. Послідовність вирішення цих завдань і визначила подальший зміст статті.

Галуззю дослідження була визначена комп'ютерна сфера, оскільки вона є однією з тих, що бурхливо розвиваються, й вірогідність перекладу відповідних текстів є достатньо високою. З іншого боку, її поняттєва структура є достатньо складною для неспеціаліста, а тому потребує попереднього засвоєння. Оскільки згадана сфера є достатньо розгалуженою, для нашого дослідження ми обрали лише один її сегмент (а саме «Структури даних»), який є важливим для роботи в межах комп'ютерної сфери загалом.

Матеріалом дослідження послужив оригінальний англомовний текст обсягом 7238 друкованих знаків, запозичений із оригінального джерела [4]. Текст було визнано репрезентативним, оскільки він є частиною авторитетного у цій галузі джерела.

Після цього ТО було перекладено українською мовою (обсяг ТП – 7791 друкований знак). Обсяг україномовного ТП (в друкованих знаках) зазвичай незначною мірою перевищує обсяг ТО англійською мовою (це пояснюється різницею в структурі згаданих мов), хоча за кількістю слів україномовний ТП є меншим: 1073 слова проти 1251 слова в англомовному ТО. Далі в обох текстах було виділено терміни та термінологічні словосполучення. Усього в кожному тексті виявлено по 115 термінологічних одиниць, які містили по 220 слів термінологічного характеру, тобто щільність текстів (відношення кількості термінів до загальної кількості слів) є досить високою – 18% в англомовному ТО та 21% в україномовному ТП. На основі відібраних термінів та термінологічних словосполучень було складено англо-український та українсько-англійський термінологічні словники.

На наступному етапі дослідження було відібрано головні поняття, на яких ґрунтуються ТО і ТП, встановлено їх ієрархію та складено поняттєву схему ТО й ТП, яку подано на рис.1, а її аналіз подано далі. Під час згаданого аналізу в усіх доцільних випадках будуть також подаватися англомовні відповідники українських термінів, що позначають поняття, показані на рис.1.

Як випливає з рис. 1, центральним у категорії «Структури даних» в даному тексті є поняття «Розв'язання проблем» (*Problem solving*), зміст якого може реалізовуватися двома способами – за допомогою алгоритмів (*algorithms*) або структур даних (*data structures*). Що стосується алгоритмів, то в програмуванні вони використовуються у вигляді методів (*methods*), функцій (*functions*) або підпрограм (*routines*). Ефективність алгоритму визначається за часом, необхідним для роботи алгоритму (*run time*), або за обсягом пам'яті, потрібним для виконання програми (*memory consumption*). При цьому, оскільки пам'ять постійно дешевшає, а її обсяг в сучасних комп'ютерах зростає, то час, необхідний для виконання програми (*run time*), є важливішим чинником, ніж обсяг пам'яті (*memory consumption*).

Як свідчить рис. 1, класифікація структур даних (*Data Structures Classification*) включає дві широкі категорії – лінійні (*linear structures*) та ієрархічні (*hierarchical structures*) структури. До лінійних відносять масиви (*arrays*), зв'язні списки (*linked lists*), стеки (*stacks*) і черги (*queues*), а до ієрархічних – дерева (*trees*), графіки (*graphs*), хіпи (*heaps*), словники (*dictionaries*) та хеш-таблиці (*hash tables*).

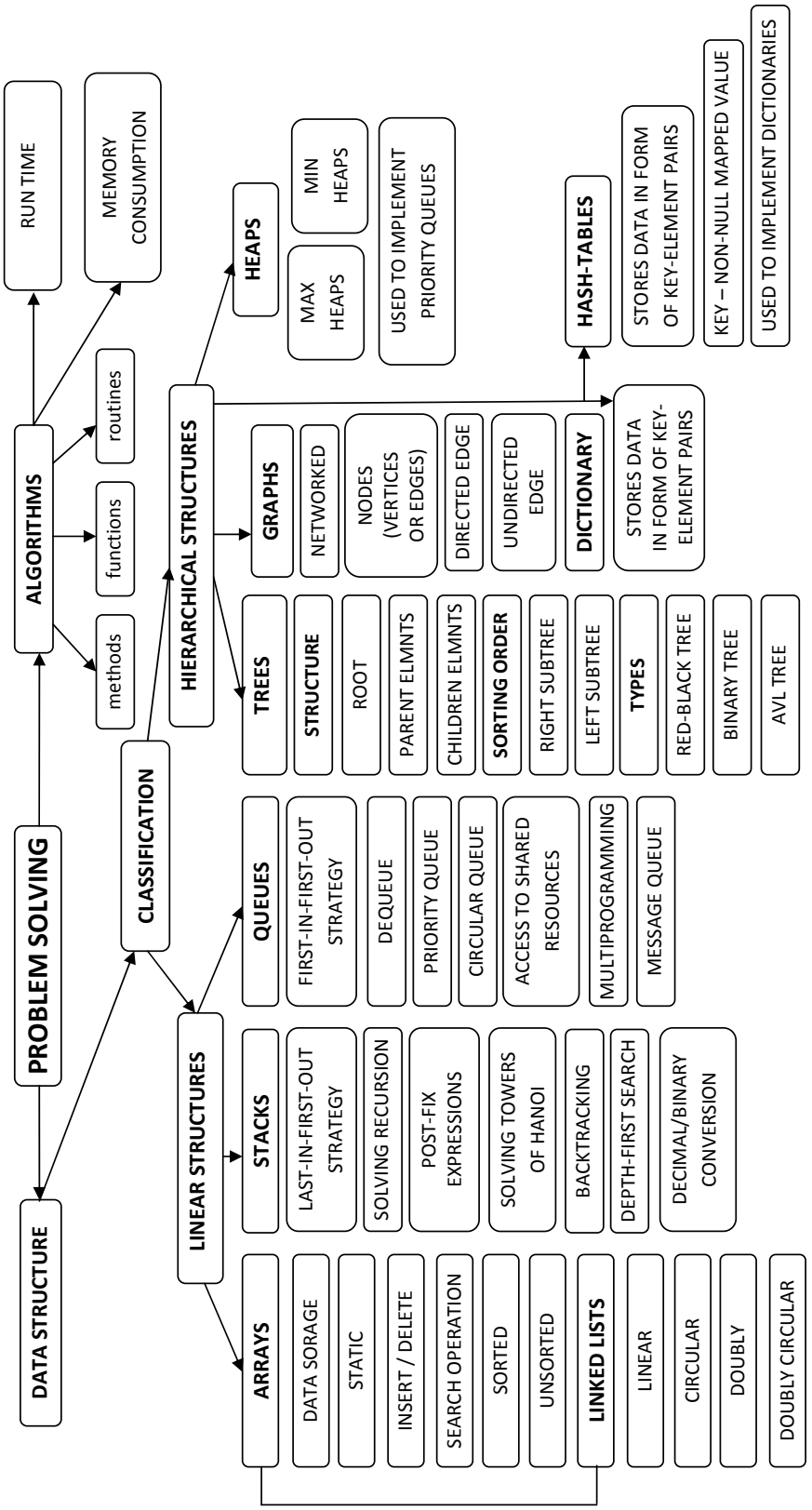

Рис. 1. Поняттєва схема до тексту (Data Structure), що служив матеріалом дослідження

Лінійні структури (*linear structures*). Перша з таких структур, що показані на рис. 1, «масив» (*array*), розглядається як структура даних, що використовується статично (*statically implemented*) у деяких мовах програмування (наприклад, *C* та *C++*), тобто обсяг масиву не може бути змінено під час його використання, хоча сучасніші мови програмування (наприклад, «Джава» (*Java*) дозволяють згадану зміну. Масиви вважаються найдоступнішою, найлегшою у використанні та найпоширенішою структурою для зберігання даних (*data storage*). Але вставка елемента (*inserting an item*) в масив або його видалення (*deleting an item*) з останнього залежить від певних чинників. Якщо потрібно вставити (*insert*) елемент у певну позицію, де вже знаходиться інший елемент, необхідно змістити всі елементи на одну позицію вправо від позиції вставки (*insertion*) нового елемента, і тільки потім вставити (*insert*) його. Час, потрібний для вставки, залежить від обсягу масиву та від позиції. Це стосується і видалення елемента (*deleting an item*). Якщо масив несортовано (*unsorted array*), то пошукова операція (*search operation*) є дорогою й може займати значний обсяг часу. Але якщо масив відсортовано (*sorted array*), то ефективність пошуку (*search performance*) значно покращується.

Наступна лінійна структура, «зв'язаний список» (*linked list*) (див. рис.1), забезпечує ефективніше керування пам'яттю (*memory management*), ніж масив. Оскільки зв'язаний список – це пам'ять, що розподіляється (*allocated memory*) під час роботи (*at run time*), то витрати пам'яті (*waste of memory*) відсутні. З іншого боку, зв'язаний список є повільнішим, ніж масив, бо відсутній прямий доступ (*direct access*) до його елементів. Його різновиди, як показано на рис. 1, включають лінійні (*linear*), кругові (*circular*), подвійні (*doubly*) та подвійно-кругові (*doubly circular*) зв'язані списки.

Ще одна лінійна структура, яку зазначено на рис. 1, «стек» (*stack*), ґрунтується на принципі «останній видаляється першим» (*last-in-first-out*), тобто першим видаляється останній збережений елемент. Стек має низку функцій, зокрема: 1) вирішення рекурсії (*solving recursion*) – рекурсивні виклики (*recursive calls*) поміщуються в стек і видаляються звідти після їх обробки (*processing*); 2) оцінка постфіксних виразів (*evaluating post-fix expressions*); 3) розв'язання головоломок типу «Ханойські вежі» (*solving Towers of Hanoi*); 4) пошук у зворотному порядку (*backtracking*); 5) пошук у глибину (*depth-first search*); 5) перетворення десяткового числа у двійкове (*converting a decimal number into a binary number*).

Останньою лінійною структурою, наведеною на рис.1, є «черга» (*queue*), що базується на принципі «перший видаляється першим» (*first-in-first-out*), тобто перший доданий до черги елемент першим і видаляється з неї. Різновидами цієї структури даних є вилучення з черги (*dequeue*), пріоритетна черга (*priority queue*) і кругова черга (*circular queue*), а варіанти її застосування (*application uses*) включають: доступ до спільних ресурсів (*access to shared resources*) (наприклад, до принтера), мультипрограмування (*multiprogramming*) та черга повідомлень (*message queue*).

Ієрархічні структури (*hierarchical structures*). Як свідчить рис. 1, в структурі дерева або деревоподібної схеми (*tree*) є корінь дерева (*root of the tree*), тобто найвищий його елемент. Крім того, кожен елемент дерева (крім кореневого) має родовий (материнський) елемент (*parent element*), а також може мати (чи не мати взагалі) один чи більше видових (дочірніх) елементів (*children elements*). Усі елементи у лівому піддереві (*left sub-tree*) стоять перед коренем у порядку сортування (*sorting order*), а ті, що містяться у правому піддереві (*right sub-tree*), стоять після кореня. Найпоширенішими варіантами деревоподібних схем є: RB-дерево (*Red-black tree*), бінарне дерево (*binary tree*), АВЛ (збалансоване)-дерево (*AVL tree*) тощо.

Наступним ієрархічним елементом (див. рис.1) є «графік» (*graph*) або «діаграма», що тлумачиться як мережева структура даних (*networked data structure*), що пов'язує набір вузлів (*nodes*), які називаються вершинами (*vertices*) або ребрами (*edges*). Ребро можна розглядати як шлях (маршрут) (*path*) або сполучну ланку (*communication link*) між двома вузлами. Ці ребра можуть бути орієнтованими (спрямованими) (*directed*) або неорієнтованими (неспрямованими) (*undirected*). Якщо шлях є спрямованим (*directed path*), то можна рухатися тільки в одному напрямку, а на неспрямованому шляху (*undirected path*) рух є можливим в обох напрямках.

Ще одним ієрархічним елементом, показаним на рис. 1, є так званий «хіп» («купа») (*heap*), що входить до категорії бінарних дерев (*binary tree*), яке зберігає набір ключів

(*collection of keys*), що мають хіпові властивості (*heap property*). Двома різними аспектами цього ієрархічного елемента структур даних є максимальний (*max heap*) та мінімальний (*min heap*) хіпи. Ознакою максимального хіпу є те, що кожен вузол (*node*) повинен бути більшим або таким самим, як його дочірні записи (*children node*). У випадку ж мінімального хіпу, кожен вузол має бути меншим або дорівнювати кожному зі своїх дочірніх записів. Хіпова структура даних зазвичай використовується для реалізації пріоритетних черг (*priority queues*).

Наступним ієрархічним елементом, репрезентованим на рис. 1, є «словник», тобто структура даних, що підтримує (*maintains*) набір елементів (*set of items*), індексованих (*indexed*) на основі ключів. Словник зберігає дані (*stores data*) у формі пар «ключ-елемент» (*key-element pairs*).

Останнім ієрархічним елементом на рис. 1 є так звана «хеш-таблиця» (*hash table*), що тлумачиться як структура даних, яка зберігає дані у вигляді пар «ключ-елемент», так само як і «словник». Ключ розглядається як ненульове значення (*non-null value*), що зіставляється (*is mapped to*) з відповідним елементом. Доступ (*access*) до згаданого елемента здійснюється на основі пов'язаного з ним ключа. Оскільки, як уже згадувалося раніше, принцип дії хеш-таблиці збігається з принципами роботи попередньо розглянутого ієрархічного елемента (словника), то згадана таблиця вважається корисною структурою даних для забезпечення функціонування останнього.

Висновки. Таким чином, ми відібрали головні поняття, на яких ґрунтуються тексти, що належать до комп'ютерної сфери, зокрема, до її сегмента «Структури даних». Сукупність згаданих понять становить предметні знання, необхідні перекладачеві для перекладу текстів, що стосуються згаданого сегмента, і мають бути засвоєні майбутніми перекладачами в процесі навчання перекладу текстів, що належать до комп'ютерної сфери. Такого ж засвоєння потребують і англомовні терміни, що означають поняття, показані на рис. 1, пов'язані з ними поняття, а також відповідні їм українські терміни. Для ефективного засвоєння як предметних знань, наведених на рис. 1, так і термінологічних одиниць для означення згаданих та пов'язаних з ними понять, необхідно розробити ефективну методику навчання, яка б забезпечила ефективне їх засвоєння, в чому і вбачаємо перспективу подальшого дослідження.

Список використаних джерел

1. Комиссаров В.Н. Современное переводоведение: [учеб. пособие] / В.Н. Комиссаров. – М.: ЭТС, 2004. – 424 с.

2. Черноватий Л.М. Методика викладання перекладу як спеціальності: Підручник для студентів вищих закладів освіти освітньо-кваліфікаційного рівня «магістр» за спеціальністю «Переклад» / Л.М. Черноватий. – Вінниця: Нова книга, 2013. – 376 с.

3. Alves F. Modelling translator's competence: Relevance and expertise under scrutiny / F. Alves, J. L. Gonçalves // Doubts and Directions in Translation Studies. – Amsterdam & Philadelphia: Benjamins, 2007. – P. 41–55.

4. Computer fundamentals [Electronic resource] / M.T. Goodrich, R. Tamassia // Algorithm Design. – NewYork | Chichester | Weinheim | Brisbane | Singapore | Toronto: John Wiley & Sons Inc., 2002. – P. 57, 61, 63,70, 77, 99, 114, 116. – Available at: http://cs-fundamentals.com/data-structures / introduction -to-data-structures.php

5. Pym A. Redefining Translation Competence in an Electronic Age. In Defence of a Minimalist Approach / A. Pym // Meta: Translators' Journal. – 2003. – Vol. 48, № 4. – P. 481–497.

References

1. Komissarov, V. N. (2004). *Sovremennoe perevodovedenie [*Modern translation]. Study guide. Moscow: ETS Publ. 424 p. (In Russian).

2. Chernovatiy, L.M. (2013). *Metodyka vykladannya perekladu yak spetsialnosti: Pidruchnyk dlya studentiv vyshchyh zakladiv osvity osvitno-kvalifikatsiynogo rivnya «magistr» za spetsialnistyu «Pereklad»* [Methodology of teaching translation as a profession: Tutorial for university students of Master's degree by Translation specialty]. Vinnytsya: Nova knyga Publ. 376 p. (In Ukrainian).

3. Alves, F., Gonçalves, J. L. (2007). Modelling translator's competence: Relevance and expertise under scrutiny. *Doubts and Directions in Translation Studies*. Amsterdam & Philadelphia: Benjamins. pp. 41-55.

4. Goodrich, M.T., Tamassia, R. (2002). Computer fundamentals [Electronic resource]. *Algorithm Design*. NewYork-Chichester-Weinheim-Brisbane– Singapore-Toronto: John Wiley & Sons Inc. pp. 57, 61, 63,70, 77, 99, 114, 116. Access mode: http://cs-fundamentals.com/data-structures/introduction-to-data-structures.php

5. Pym, A. (2003) Redefining Translation Competence in an Electronic Age. In Defence of a Minimalist Approach. *Meta: Translators' Journal*. 2003, Vol. 48, No 4. pp. 481-497.

Предложено описание процесса составления понятийной схемы и отбора англо- и украиноязычной терминологии (отрасль – «Компьютерные науки», тема «Структуры данных») для формирования предметной и терминологической составляющих профессиональной компетентности переводчика как этапа разработки экспериментальной методики обучения отраслевому переводу.

Ключевые слова: обучение отраслевому переводу, компьютерные науки, структуры данных, профессиональная компетентность переводчика, предметная компетентность, терминологическая компетентность, английский и украинский языки, понятийная схема.

The articles describes the process of drawing up a notional scheme and the selection of English and Ukrainian terminology (sphere – Computer Sciences, topic – Data Structures) for the TC subject-matter and terminological components acquisition as a stage in the development of an experimental non-fiction translator teaching technology.

Key words: teaching non-fiction translation, Computer Sciences, Data Structures, professional translator's competence, subject-matter competence, terminological competence, English and Ukrainian languages, notional scheme.

Одержано 28.01.2016.