

## АДАПТАЦИЯ АНГЛОЯЗЫЧНЫХ КОМПЬЮТЕРНЫХ ТЕКСТОВ ПРИ ПЕРЕВОДЕ НА РУССКИЙ ЯЗЫК

*Л.В. Стахова (Санкт-Петербург, Россия)*

*В докладе рассматривается проблема адекватной передачи единиц англоязычной компьютерной терминосистемы при переводе на русский язык.*

**Ключевые слова:** компьютерный текст, компьютерный термин, перевод

## RUSSIAN ADAPTATION OF ENGLISH COMPUTER LITERATURE TEXTS

*L. Stakhova (Saint-Petersburgh, Russia)*

*The study addresses the issue of computer term rendering while translating computer literature texts from English into Russian.*

**Keywords:** a computer literature text, a computer term, translation

Грамотный перевод текстов, обслуживающих ту или иную профессиональную сферу, требует тщательного предварительного изучения специфики данного направления человеческой деятельности. Освоение переводчиком области профессиональных знаний особенно значимо при переводе компьютерных текстов, содержащих мощный пласт профессионализмов.

Во-первых, данная лексика представляется собой группу динамично изменяющихся языковых единиц. Процесс появления новых слов и переход существующих в разряд устаревшей лексики определяется исключительно экстралингвистическими факторами, такими как модернизация в самой отрасли, появление новых информационных технологий и компьютерного оборудования, требующих своего обозначения в языке.

Во-вторых, в компьютерном подязыке можно выделить большую группу единиц, которые появились как семантические новообразования, так называемый процесс терминологизации [Кондратюкова, 2013: 279].

В-третьих, лексическая подсистема компьютерного языка включает самые разнообразные элементы почти всех языковых уровней.

В-четвертых, терминологическая система компьютерных технологий неоднородна по структуре, т.к. статус пользователей данного подязыка варьируется от профессионалов до, например, пользователей социальных сетей, чат-коммуникаций, геймеров и др. Каждая социальная группа использует свой определенных набор языковых средств.

Все это способствует выделению подязыка компьютерных технологий как специфической терминосистемы, требующей неоднозначного комплексного переводческого решения.

В данной работе рассматривается только один аспект функционирования компьютерного подязыка, а именно англоязычные лексические единицы, выявленные в тексте книги Бланшет Ж., Саммерфилд М. «Qt 4. Программирование GUI на C++», т.е. лексика, обслуживающая сферу деятельности IT – специалистов.

Для того чтобы определить возможные способы адаптации компьютерных текстов при переводе, необходимо, в первую очередь, проанализировать особенности создания профессионализмов IT – технологий в языке оригинала.

Анализ материала исследования показывает, что лексика компьютерного подязыка образуется как с помощью традиционных (семантические новообразования, аббревиация, конверсия, словосложение, аффиксация и др.), так и нетрадиционных способов словообразования.

Как уже упоминалось выше, наиболее активным является семантическое терминообразование. В качестве примера можно привести единицу «string» со значением «веревка», «завязка», «струна». На основе метафорического переноса в языках программирования слово приобрело новое значение: а) тип данных, позволяющий хранить и обрабатывать последовательности символов, поэтому иногда её называют символьной, или текстовой, строкой; б) упорядоченная последовательность значений данных: «строка», «цепочка», «последовательность».

Every named child of the QObject (that is, for which QObject::objectName() is not an empty **string**) is by default available as a property of the QScript wrapper object.

Также активно функционируют следующие способы словообразования.

- Аббревиация: API – Application Programming Interface («интерфейс прикладного программирования»). Данная единица появилась в результате сложения первых букв выражения Application Programming Interface.

The problem is that this method reveals too much of the internal **API** and gives script programmers access to application internals which should not be exposed.

- Конверсия: существительное «overload» – «перезагрузка», произошедшее из глагола «перезагрузить».

There are three **overloads** of connect(), each with a corresponding disconnect() **overload**;

- Словосложение: Run-time – «время выполнения программ». Единица образована сложением двух основ: run – «выполнение» и time – «время»

All classes which derive from QObject can be introspected and can provide the information needed by the scripting engine at **run-time**.

- Аффикация: Interface – «интерфейс» (в языках программирования – видимая пользователю, в отличие от реализации, часть описания (функции, модуля, класса), определяющая способ их использования). Термин образован путем использования приставки inter-.

The simplest way to solve this is to only use QObject\* and QWidget\* in the method signatures of your scripting **interface**.

Специфичность лексики компьютерного подязыка в целом и характер исследуемого текста обусловили появление сложных единиц, образованных путем комбинации нескольких способов словообразования.

- Аббревиация в сочетании со словосложением: QScriptEngine, QObject

Первая единица образована путем сложения начальной буквы Q от названия Qt и двух основ – Script и Engine в первом случае и Q и слова Object – во втором.

When a QObject is passed to the **QScriptEngine::newQObject()** function, a Qt Script wrapper object is created that can be used to make the QObject's signals, slots, properties, and child objects available to scripts.

- Синтаксический способ в сочетании с аббревиацией и словосложением: Microsoft's JScript – для образования этого словосочетания использовалась модель: существительное в притяжательном падеже Microsoft's сочетается с существительным Jscript. Единица JScript в свою очередь образована с помощью сложения начальной буквы J от названия Java и слова Script.

**Microsoft's JScript** and Netscape's JavaScript are also based on the ECMAScript standard.

Интересным вариантом семантического словообразования может служить пример использования названия марки кофе Java для названия языка программирования.

Разнообразие представленных выше элементов терминосистемы компьютерного подязыка требует своей адекватной передачи при переводе с английского на русский. В настоящее время не существует однозначных решений данной проблемы. Представленный далее анализ вариантов перевода некоторых единиц может послужить лишь отправной точкой в исследованиях данного направления переводоведения.

Достаточно часто при переводе используется подбор словарного соответствия.

The script function can, for example, be a function that the **user** has typed in, or one that you have read from a file – Скриптовая функция может, например, быть функцией, которую **пользователь** напечатал сам или считал из файла.

В данном примере существительное «user», общеупотребительное значение которого «пользователь», переводится словарным соответствием.

**Default Conversion from Qt Script to C++ – По умолчанию** переход от Qt Script к C++.

В словаре компьютерной лексики, существительное «default» обозначает «по умолчанию», и в данном предложении передается как словарное соответствие.

Переводчики подобных текстов нередко прибегают к использованию трансплантации, приема при котором сохраняется иноязычная форма написания единицы в тексте перевода [Кабакчи, 1998: 64].

**Qt Script** is based on the **ECMAScript** scripting language, as defined in standard ECMA-262 – **Qt Script** основан на **ECMAScript** – скриптовом языке, определенном в стандарте ECMA-262.

Единицы Qt Script, ECMAScript передаются на русский язык без каких-либо изменений.

Одним из оптимальных способов, используемых при передаче подобных текстов, является калькирование.

A **nested function** can be used to «capture» the execution context in which a nested function object is created – **Вложенная функция** может использоваться, чтобы «захватить» контекст выполнения, в котором создается объект вложенной функции.

В данном примере представлена покомпонентная передача термина на русский язык.

Можно наблюдать использование генерализации в некоторых примерах.

**Hybrid C++/script:** C++ application code connects a signal to a script function – **Схема C++/script:** сигнал из кода приложения привязывается к скриптовой функции.

В словаре единица hybrid обозначает «гибридная схема», при переводе компьютерных текстов частное понятие гибридная схема заменяется общим – «схема».

В следующем примере представлен случай конкретизации.

For example, a script can define a function that will **handle** the QLineEdit:returnPressed() signal, and then connect that signal to the script function – Например, скрипт может описывать функцию, которая будет **обрабатывать** сигнал QLineEdit: returnPressed, и затем присоединять этот сигнал к скриптовой функции.

Handle, что в переводе на русский язык обозначает «управлять», «обращаться», «трогать», приобретает конкретное значение и в компьютерной лексике переводится как «обрабатывать».

Иногда для адекватной передачи значения исходной единицы приходится прибегать к добавлению.

By default, the script engine does not take **ownership** of the QObject that is passed to QScriptEngine:newQObject() – По умолчанию скриптовый движок не использует **режим владения** QObject, который передается в QScriptEngine: newQObject ().

При переводе добавляется единица «режим»: ownership – «режим владения».

Иногда переводчику приходится прибегать к целостному преобразованию исходной единицы.

You can specify a particular overload by **using array-style property access with the normalized signature of the C++ function as the property name.** – Вы можете определить специальную перегрузку, **используя доступ к свойству как к массиву с нормализацией сигнатуры метода функции C++ как имя свойства.**

Часто применяются транскрибирование и транслитерирование единиц.

Qt Script adapts Qt's central Signals and **Slots** feature for **scripting.** – Qt Script дает возможность использовать базовые понятия Qt – сигналы и **слоты** – в **скриптах.**

Специфичность лексических единиц оригинального текста требует творческого подхода к процессу их перевода. Так, возможно использование сразу нескольких переводческих трансформаций для передачи одной единицы.

Например, словосочетание «clicked signal» было передано на русский язык при помощи калькирования, но в связи с особенностями сферы употребления, слово «clicked» было трансплантировано, и выражение «clicked signal» переводится, как «сигнал clicked».

If you have a push button in a form, you typically want to do something involving the form in response to the button's **clicked signal.** – Если у Вас есть кнопка на форме, вы обычно хотите сделать что-то, позволяющее отреагировать на **сигнал clicked** этой кнопки.

Калькированием с частичной трансплантацией передается и единица C++ objects.

This mode is appropriate when, for example, you are wrapping **C++ objects** that are part of your application's core – Этот способ является подходящим, когда, например, вы обортываете **C++ объекты**, которые являются частью ядра приложения.

Трансплантация может сопровождаться добавлением.

**Microsoft's Jscript, and Netscape's JavaScript** are also based on the ECMAScript standard – **Jscript компании Microsoft и JavaScript компании Netscape** также основаны на стандарте ECMAScript.

В переводе используются транспланты Microsoft's Jscript, и Netscape's JavaScript с добавлением слова «компания».

Рассмотренные в работе способы передачи лексики IT-сферы демонстрируют отдельные аспекты глобальной проблемы адаптации англоязычных компьютерных текстов при переводе на русский язык. Остальные вопросы, такие как, например, использование графических средств, и многие другие ждут своей дальнейшей разработки.

## Список литературы

- [1] *Кабакчи В.В.* Основы англоязычной межкультурной коммуникации. – СПб.: РГПУ им. А. Герцена, 1998.
- [2] *Кондратьюкова Л.К.* Статус термина в системе языка // Омские социально-гуманитарные чтения – 2013. Материалы VI Международной научно-практической конференции. – Омск: Издательство Омского государственного технического университета, 2013.