Сафонов Владимир Олегович

КПП: КОЛЛЕКЦИЯ ПРАВИЛЬНЫХ ПЕРЕВОДОВ

ВВЕДЕНИЕ

Мотивацией моей статьи послужила очередная ошибка студентов в докладе на семинаре по компиляторам для студентов 5 курса [1]. Сам доклад был интересен и добросовестно подготовлен – об организации таблиц в Microsoft Phoenix [2], новом инструментарии Microsoft для разработки компиляторов, первыми академическими пользователями которого мы – группа моих учеников и я являемся с 2003 г.

Однако ошибка в переводе термина – перевод *evaluation* как *оценка* вместо *вычисление* (имелось в виду вычисление некоторого выражения), сделанная авторами доклада – способными студентами, настолько задела меня за живое, что вызвала мой пятиминутный «монолог» на семинаре, в котором я объяснил студентам некоторые типичные ошибки при переводе на русский язык профессиональных английских терминов из области информационных технологий (ИТ). К сожалению, вина в этих ошибках – не только и не столько самих студентов. Неправильными переводами терминов, искажающими содержание статей и книг, грешат многие издания.

Анализируя эту ситуацию, я пришел к выводу, что многим студентам и специалистам был бы интересен и полезен словарь рекомендуемых *правильных* переводов на русский язык англоязычных терминов из области ИТ, – в особенности, таких, для которых распространение получили *неверные* переводы. По моему замыслу, словарь должен быть доступен через Web для просмотра, изменений и дополнений.

Считаю правильность профессиональной англоязычной терминологии (и вообще, хорошее знание английского языка) одним из важнейших качеств специалистов по ИТ, которых мы с вами готовим.

Поэтому пусть данная статья послужит отправной точкой для конструктивной дискуссии на эту тему и для создания нашими общими усилиями словаря правильных и типичных неправильных переводов (во избежание повторения ошибок) англоязычных терминов в области ИТ, который будем называть *КПП – Коллекцией Правильных Переводов*.

Пусть каждый перевод английского термина из области ИТ на русский язык (разумеется, неофициально и на добровольной основе) *«проходит через наш КПП»*. Надеюсь, что благодаря этому переводимый текст станет более понятным и полезным читателю, а не превратится в почти бессмысленный набор слов, механически повторяемых студентами на докладе или при ответе на экзамене, как это, к сожалению, иногда случается с неудачными переводами профессиональных текстов.

Буду рад вашим откликам на эту статью и материалам – новым статьям нашего словаря КПП, которые приглашаю уважаемых читателей присылать мне по электронной почте: *v_o_safonov@mail.ru*.

Когда словарь КПП станет достаточно большим, он будет опубликован на сайте нашей лаборатории Java-технологии Санкт-Петербургского государственного университета [3].

ENGLISH IS YOUR TOOL

Этим девизом (*«Английский язык – Ваш инструмент»*) я постоянно убеждаю своих студентов и аспирантов в том, что специалистам необходимо в настоящее время не только хорошее знание английского языка, но и практическое владение им как одним из инструментов в области ИТ, столь же важным, как Java-технология [4, 5] и платформа .NET [6], которыми я занимаюсь.

К сожалению, приходится сделать вывод, что до сих пор одной из основных причин недостаточного мирового признания результатов и заслуг отечественной науки (в частности, в области ИТ) является недостаточно хорошее знание английского языка российскими специалистами. Из-за этого англоязычные статьи и доклады наших авторов не проходят отбор в журналы или на конференции. Более того, иногда, к сожалению, собственное неуверенное владение английским языком вообще останавливает наших специалистов, пресекая их естественное желание написать англоязычную статью о своих результатах в какой-либо известный журнал, чтобы сделать их более популярными в мире.

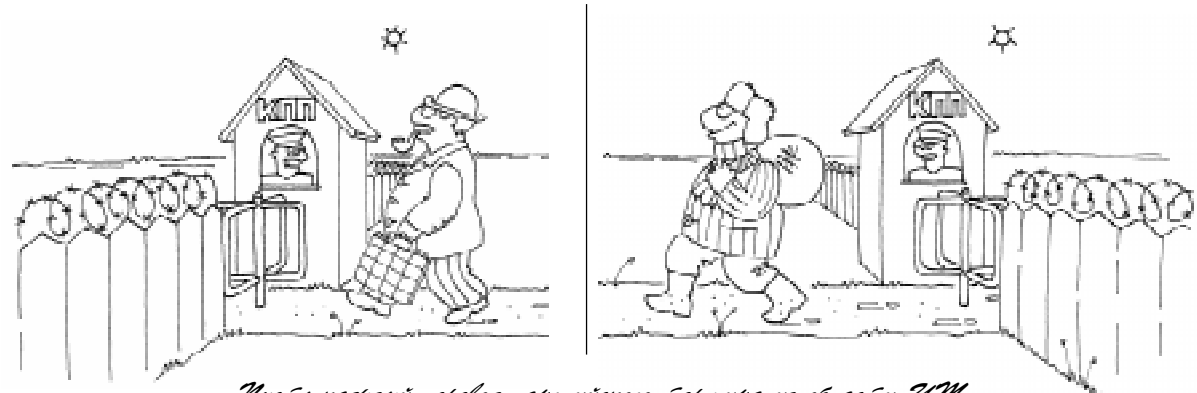
Немного о моем опыте написания и публикации статей и книг на английском языке.

С 2003 г. являюсь постоянным автором известного американского журнала «.NET Developer's Journal» – журнала разработчиков программ для платформы .NET. Опубликовал в нем серию статей [7–11] об аспектно-ориентированном программировании и нашем инструменте Aspect.NET, используемом в 17 странах мира, о своих преподавательских проектах SPBU.NET и TrustSPBU.NET (все эти проекты поддержаны Microsoft Research). В 2007 г. написал книгу [12] на английском языке об использовании аспектно-ориентированного программирования для разработки надежных и безопасных программ, которая в 2008 г. выйдет в США в издательстве John Wiley & Sons.

Постоянно совершенствую свой профессиональный английский язык, что рекомендую делать и всем своим ученикам.

Из источников, которые постоянно использую, рекомендую толковый словарь американского английского языка Longman [13], опубликованный как в виде книги, так и на компакт-диске.

На мой взгляд, правильность профессионального английского языка начинается с терминологии – с правильности, адекватности, ясности перевода англоязычной терминологии на русский язык и, наоборот, русскоязычных профессиональных терминов на английский. К сожалению, в данной области наблюдаются большие проблемы. Вот некоторые тенденции и причины ошибок в переводе с английского на русский профессиональных текстов и терминов из области ИТ:



Пусть каждый перевод английского термина из области ИТ на русский язык ... «проходит через наш КПП».

• **«Буквальный» перевод терминов** с использованием приемлемого для *повседневной лексики* варианта перевода, не подходящего, однако, по смыслу к области ИТ. Например, неверный перевод термина ***reflection*** как *отражение* (правильный вариант – ***рефлексия***; необходимые пояснения даны ниже).

Попытки использовать совпадающий по звучанию аналог английского термина...

• **Попытки использовать совпадающий по звучанию аналог английского термина**, уже имеющий эквивалент в русском языке, который по смыслу, однако, имеет *другое* значение. Пример: неверный перевод термина ***operator*** как *оператор* (правильный вариант – ***операция*** или ***знак операции***; пояснения приведены ниже).

• **Использование некачественных программ-переводчиков**. Предпочту не приводить конкретных названий, однако, к сожалению, практически все программы-переводчики, проанализированные мной или используемые моими студентами, генерируют неадекватный, некачественный, неуклюжий перевод. Рекомендую при переводе ими вообще не пользоваться. В дальнейшем, надеюсь, будут, наконец, разработаны программы-переводчики профессиональных текстов с приемлемым качеством перевода.

НАЧНЕМ СЛОВАРЬ КПП

Ниже приводятся примеры некоторых наиболее типичных ошибок перевода терминов из области ИТ с английского языка на русский – будущие статьи словаря КПП.

Жирным курсивом выделены сами термины и их рекомендуемый правильный перевод, обычным (нежирным) курсивом – неверные переводы терминов, которые, к сожалению, достаточно широко распространились в профессиональных русскоязычных текстах – книгах, статьях, диссертациях. В необходимых случаях даны пояснения. Большинство рекомендуемых мной переводов основаны на многолетней традиции отечественного программирования, которым я и предлагаю следовать.

В дополнение к данному материалу, в моих публикациях прошлых лет [5, 14] в приложениях приведены глоссарии – рекомендуемые мной переводы англоязычных терминов на русский язык, соответствующих контексту данных работ: в книге [14] – по языку CLU, абстрактным типам данных, инженерии программ; в книге [5] – по Java-технологии.

Evaluation **(of expression)** – ***вычисление*** **(выражения)**, а не *оценка*. Перевод *evaluation* как *оценка* (или, лучше, *исследование*) уместен в несколько другом контексте: например: ***problem evaluation – исследование проблемы***.

Operator **(e.g., «+»)** **–** ***операция*** **либо** ***знак операции*** **(например, сложения)**, а не *оператор*. Российские специалисты привыкли за десятки лет называть «+» именно ***операцией***, а не *оператором*, но, увы, многие современные некачественные переводы пытаются сломать эту традицию.

Statement **(executable construct of a programming language, e.g., SWITCH)** **–** ***оператор*** **(исполняемая конструкция языка программирования, например, оператор SWITCH)**, а не *утверждение:* операторы ничего не *утверждают,* а управляют выполнением программ.

Precedence **(of operators)** **–** ***приоритет*** **(операций)**, а не *предшествование*. Например, **операция** (*а не оператор*) «*» имеет больший **приоритет** (*а не предшествование!)*, чем операция «+». В частности, укоренившиеся из-за неточного перевода русскоязычные термины *отношения предшествования* **(*precedence relation*)** и *анализатор предшествования* **(*precedence parser*)** следовало бы заменить на более длинные, но более правильные: ***соотношения приоритетов операций*** **и** ***синтаксический***

***анализатор на основе (с учетом) приоритетов операций*.**

Applied occurrence* (of identifier)** – в компиляторах: ***использующее вхождение или просто ***использование*** **(идентификатора)**, а не *прикладная реализация* (как в некоторых переводах книг по компиляторам)

Information management – управление информацией, а не *информационный менеджмент*. Приведенный (увы, распространенный) неточный перевод данного термина создает ложное впечатление, что термин имеет отношение к менеджменту, то есть управлению коллективами и организациями. Однако в англоязычной литературе он употребляется именно в значении ***управление информацией*** – например, к данной области относится большинство разработок фирмы EMC, широко известной своими решениями для хранения данных; многие продукты Microsoft (Office, SharePoint и др.), которые предназначены именно для *управления* различного рода информацией, а не для менеджмента (хотя, разумеется, используются и менеджерами).

Аналогично: ***knowledge management* – управление знаниями.**

Reflection* (in Java and .NET) – *рефлексия* (в Java и .NET)**, а не *отражение*. Термин ***рефлексия весьма удачно выбран авторами Java и .NET. По своей природе он метафоричен, носит общефилософский характер (рефлексия – рассуждения человека о самом себе, самоанализ). В литературе по Java и .NET термин ***рефлексия*** обозначает анализ типов во время выполнения программы. Рефлексия позволяет, например, проанализировать структуру любого класса (в том числе – полученного по сети и динамически загруженного в виртуальную машину), его методы и поля, а также вызвать любой метод данного класса без использования его исходного текста, только на основе информации, полученной в результате рефлексии. Использовать для обозначения подобных возможностей современных систем программирования термин *отражение* ошибочно: программа на Java «рефлексирует» (анализирует структуру своих типов, то есть как бы «рассуждает о самой себе»), но ничего не «отражает», хотя отражение и является одним из обыденных значений слова *reflection*.

Control – управление, а не *контроль*. Например, ***access control list (ACL)* – список для управления доступом** (а не для *контроля* доступа). Пользуясь случаем, не могу не высказаться относительно неверного использования термина *control* как английского перевода термина «проверка, контроль» (например, билетов). Например, в одном из известных всем петербуржцам и всему миру наших художественных музеев на видном месте при входе рядом с пожилыми сотрудницами музея – *контролерами* входных билетов – красуется крупная надпись *CONTROL*, вызывающая у иностранцев легкую улыбку. Мне неоднократно приходилось с шутливыми извинениями комментировать им эту ошибку перевода: «It looks like this lady controls the whole building» – «Похоже, что эта дама управляет всем этим зданием». Возможный правильный вариант – ***checking (tickets)***, либо более уместная в данном случае надпись ***entrance*** (вход). Привожу этот пример из жизни только потому, что еще раз хочу подчеркнуть правильные соответствия близких по смыслу, но различных терминов:
control – управление,
***checking – контроль* (билетов, документов),**
***testing – тестирование* (*проверка*) программы.**

***Member* (of a type) – *элемент* (типа) – поле или метод**. Перевод *member* как *член* не рекомендуется из соображений благозвучия, хотя он, к сожалению, очень распространен.

Swapping – откачка и подкачка* (а не *перестановка и т. д.). Данный термин относится к области операционных систем и означает откачку и подкачку страниц – обмен информацией между оперативной памятью и образом виртуальной памяти на диске. К сожалению, мне неоднократно приходилось сталкиваться с различными неверными переводами данного термина. Рекомендованный мной перевод десятки лет используется в отечественной литературе по ОС.

Parsing – синтаксический анализ (рекомендую избегать использования термина *парсинг* и, соответственно, *парсер*). Данный термин относится к области компиляторов и обозначает фазу синтаксического анализа, то есть, преобразование исходного текста программы (или потока лексем) в дерево вывода. Термин *парсинг*, который пытаются нам привить некачественные переводы, мне представляется не вполне благозвучным (напоминает *пирсинг*).

Type safety – безопасность использования типов (или ***обработки типов***), а не *типовая безопасность*. Данный термин весьма распространен в современном программировании благодаря популярности платформ Java и .NET. Считаю, что перевод *типовая безопасность* звучит по-русски неуклюже, поэтому рекомендую приведенные мной варианты перевода.

Соответственно, ***type checking*** рекомендую переводить как ***контроль типов***.

Приведенные в статье переводы терминов – это только начало.

Читатели приглашаются к активному участию в составлении и обсуждении словаря КПП.

Надеюсь, что наш словарь КПП будет расти, совершенствоваться и, самое главное, широко использоваться молодыми российскими программистами, что послужит улучшению ситуации в области профессионального английского языка и дальнейшему совершенствованию отечественного высшего образования в области ИТ.

Литература

1. *Safonov V.O.* Compiler Development. Graduate university seminar. http://www.msdnaa.net/curriculum/?id=6244
2. Web-сайт Microsoft Phoenix. http://research.microsoft.com/phoenix
3. Web-сайт лаборатории Java-технологии НИИ математики и механики им. акад. В.И. Смирнова математико-механического факультета СПбГУ. *http://polyhimnie.math.spbu.ru*
4. *Сафонов В.О.* Java-технология: история, состояние и перспективы // Компьютерные инструменты в образовании, 2003, № 2.
5. *Сафонов В.О.* Введение в Java-технологию. СПб.: Наука, 2002.
6. *Сафонов В.О.* Платформа Microsoft.NET: принципы, возможности, перспективы // Компьютерные инструменты в образовании, 2004, № 4.
7. *Safonov V.O.* Aspect.NET – a new approach to aspect-oriented programming. .NET Developers Journal, April 2003.
8. *Safonov V.O.* Aspect.NET – concepts and architecture. .NET Developers Journal, October 2004.
9. *Safonov V.O., Grigoriev D.A.* Aspect.NET – aspect-oriented programming for Microsoft.NET in practice. .NET Developers Journal, July 2005.
10. *Safonov V.O.* SPBU.NET – principles and experience of teaching Microsoft.NET, compilers, software engineering and OS. .NET Developer's Journal, February 2006.
11. *Safonov V.O.* TrustSPBU.NET: Extending university courses on .NET, compilers, software engineering and OS by trustworthy computing content. .NET Developer's Journal, March 2007.
12. *Safonov V.O.* Using Aspect-Oriented Programming for Trustworthy Software Development. ISBN 9780470138175. – John Wiley &Sons, 2008, to be published.
13. Longman Dictionary of American English, Pearson, Longman, 2005.
14. *Сафонов В.О.* Языки и методы программирования в системе Эльбрус. М.: Наука, 1989.

Сафонов Владимир Олегович,
доктор технических наук,
профессор кафедры информатики
СПбГУ, руководитель лаборатории
Java-технологии.