Мысли об организации перевода документации по Qt

Исходная документация
Необходимо иметь исходную документацию по каждой версии. Она должна являться эталоном для перевода.
Почему предлагаю именно хранить на сервере? В принципе, есть вариант где-то дать ссылку, например, на архив с официальной документацией. Но, во-первых, я такого архива на FTP Nokia не нашёл, а во-вторых, когда ближе к телу, спокойнее :). Кроме того, даже документация одной и той же версии (например, 4.7.4) может быть разной - в частности, разное HTML-обрамление. Поэтому лучше, если все переводчики будут работать с однозначно одинаковой документацией.
С одной стороны, хорошо, если в этой документации будет всегда полный набор файлов. С другой, возможно, хорошо оставлять там только то, что нужно перевести. Предварительный вывод – нужны все файлы исходной документации (вероятно, архив) и какого-то рода список того, что нужно переводить.
В именах архивов должна присутствовать в том или ином виде полная версия документации, например, не qt-doc-4.7-en.7z, а qt-doc-4.7.4-en.7z. В принципе, сейчас в папке /qtrtt/tm/etalon так и есть. Но мне ещё кажется правильным видеть в названии файла, что это документация, и что она английская.

Конечная документация
Просто архивы сгенерированной документации по версиям. Нет смысла хранить исходные файлы, которые переведены, в каталоге target. Архивы генерируются и обновляются только ответственным за объединение. У других членов группы есть к ним доступ только на чтение.
Поскольку конечная версия документации доступна через сайт, то указание в имени архивов полной версии - по усмотрению администратора сайта. Но, думаю, будет красиво, если стиль именования будет одинаковый.

Рабочая документация
Вряд ли сами файлы, которые в работе, нужно перемещать в отдельный каталог. Думаю, нужен просто список файлов в работе, причём с указанием того, кто этот файл взял в работу. На перевод конкретного файла даётся какое-то время, например, от месяца до цикла (3 месяца). Если за этот срок файл не был переведен, то он снимается с работы и может быть снова взят в работу другим переводчиком. Естественно, желательно согласовать этот шаг с предыдущим переводчиком.

Списки
Вариант 1 - мысли.
Нам нужно что-то вроде таблицы. По вертикали – список файлов. По горизонтали – версии. В ячейках помечаем, что файл для определённой версии взят в работу. Пометка означает лишь то, что файл в работе. Снимать пометку переводчик может только в том случае, если он отказывается от перевода файла. Список генерируется заново автоматически при создании новой версии документации. Список может быть доступен по SVN или git, например.
Подумал, решил, что таблица – плохо. Возникают конфликты в SVN и git и непонятно, что делать, если файл в одной версии переведён, а в другой – нет, ячейка-то для него в любом случае есть. Можно, конечно, ее чем-нибудь заполнять….
Вариант 2 - мысли.

Лучше последовательный список для всех версий в одном файле, либо разные файлы для разных версий. Допустим, что есть отдельные списки файлов для каждой версии Qt. Просто обычные списки. Тогда разработчик может ставить свой ник напротив тех файлов, которые берёт в работу. Удобно при одной версии. Не очень удобно при нескольких. В этом случае, если переводчик хочет поддерживать все версии, придётся в каждом списке ставить пометку для выбранных файлов, что геморно. Можно было бы, конечно, сделать ещё один список – всех версий, в который попали бы файлы, не переведённые хотя бы в одной версии. Тогда если переводчик хочет перевести файл для всех версий, он ставит пометку в нём. Но в этом случае есть две проблемы: первая, когда переводчик берет файл одной версии, в общем списке нет пометки, и это значит, что переводчик, желающий перевести все, должен пробежаться по всем спискам; вторая ситуация - обратная первой.
Вариант 3 - наиболее удачный.
Ещё один вариант – в список попадают все не переведённые файлы всех версий документации с пометкой, для каких версий файлы не переведены. Переводчик в строке с именем файла в свободном стиле указывает те версии, для которых он будет делать перевод.
В любом случае, не хочется, чтобы два переводчика переводили один и тот же файл в разных версиях – при слиянии будет много конфликтов. Поэтому будет логичным, что резервирование файла – это резервирование файла для всех версий. По окончании перевода при очередном слиянии из списка удалятся те версии, что реально переведены и останутся только непереведённые.
Прихожу к выводу, что нужно иметь один файл, в котором будет список всех непереведённых хоть в какой-либо из поддерживаемых версий файлов. В каждой строке указаны имя файла и в скобках список версий, для которых он не переведён. Дополнительно, если файл не переведён ни в одной версии, его можно помечать флагом ALL, если переведён в основной версии, но не переведён хотя бы в одной из дополнительных – AUX, если полностью переведён в одной из дополнительных, но требует доперевода в основной – BASE. Если получится с дополнительными метками, то можно оставить только их, но я бы всё-таки указывал и список версий.

Управление списками
После генерации перевода в каталоге omegat появляется файл project_stats.txt, в котором указана статистика перевода. Каким-то образом (пока не знаю, каким, видимо, придётся что-то своё сделать. upd: уже сделано) формируется список, в котором в алфавитном порядке перечислены файлы, которые не переведены хотя бы в одной из поддерживаемых версий. После имени файла в скобках указаны номера версий, в которых файл не переведён. Переводчик при взятии файла в перевод в свободном стиле указывает в строке с именем файла, что он (либо кто-то ещё) взял его в работу. Например, можно просто указывать свой ник.
После слияния переводов и генерации новых версий документации из старого списка исключаются те файлы, которые полностью переведены.
В дальнейшем работа идёт по этому новому списку.
Файл со списком можно оставить на FTP - не так уж часто к нему и обращаются, а можно поместить его в репозиторий, тогда можно будет отслеживать, когда что менялось: кто что брал в работу, какие были изменения при последних слиянии и генерации. Также в репозиторий можно будет поместить глоссарий и файл проекта.

Версии
Приоритет 1. Определяется одна основная версия, над которой работает группа. Эта версия является наиболее приоритетной для перевода. На данный момент это 4.7.4.

Приоритет 2. На втором уровне стоят какие-то (по возможности, последние) подверсии в каждой из версий (например, подверсия 4.3.5 для версии 4.3, подверсия 4.4.3 для версии 4.4 и т.д.). Кстати, с учётом приоритета 1 можно генерировать документацию и на более новые версии, например, 4.8.2 на данный момент. Но поскольку старшие версии выходят более часто, то можно не заниматься поддержкой их перевода, пока очередная старшая версия не станет приоритетом 1.
Приоритет 3. На третьем уровне – другие версии и подверсии в пределах текущей версии первого уровня. Например, если сейчас базовой версией является 4.7.4, то можно переводить любые версии 4.x.x, если это необходимо конкретному переводчику.
Приоритет 4. Такая структура может соблюдаться и для других версий первого уровня (1.x.x, 2.x.x, 3.x.x и т.д.). Это будет четвёртый уровень приоритетности.

Список поддерживаемых версий
Список поддерживаемых группой версий и подверсий выкладывается на сайте. Группа концентрирует усилия на переводе версий приоритетов 1 и 2.
После обновления ПП для сайта генерируется (почему, кстати, на сайте какая-то своя документация, отличная от той, что лежит на FTP?) документация по тем версиям, которые были в списке на момент обновления ПП.
После этого список версий может быть откорректирован:

- возможно, в каждой версии второго уровня (4.3, 4.4, 4.5...) выбирается последняя из существующих подверсий;
- возможно, меняется версия приоритета 1.

Не знаю, что может оказать влияние на смену списка. alex977 вполне резонно предложил не брать в перевод новую версию второго уровня, пока не появятся подверсии x.x.2/x.x.3. Формализовать свои мысли в этой части я не могу.

Память переводов
Ниже речь идёт о переводе всех версий документации одного первого уровня (например, все версии 4.x.x). Для них на первичный момент существует память переводов.

1. Она скачивается разработчиками и помещается в подкаталог omegat каталога с проектом под именем project_save.tmx.
2. По окончании цикла каждый переводчик генерирует разностную память переводов по сравнению с последней зафиксированной памятью переводов с помощью qtmxtools.
3. Ответственный за объединение мержит эти диффы в последнюю зафиксированную версию памяти переводов с помощью qtmxtools.
4. Полученная на шаге 3 память переводов копируется в подкаталог omegat каталога с проектом перевода всех версий документации данного первого уровня (например, 4.3.5, 4.4.3, 4.5.3, 4.6.4, 4.7.4, 4.8.2) под именем project_save.tmx.
5. Далее создаются переведённые файлы этих версий с использованием этой памяти переводов. Перевод должен быть осуществлён за один раз, т.е. все версии должны быть в папке source.
   Правда, в этом случае может возникнуть проблема с нехваткой памяти. При запуске Java можно указать ключ -Xmx1500M, где 1500M - размер выделяемой памяти. С памятью переводов от июля 2012 одна версия документации отнимает около 250 Мб памяти. Хотя, может, эта цифра от памяти перевода не зависит, не знаю.
   Либо можно осуществить перевод за несколько итераций, но тогда придётся объединять файлы с шага 6.
6. В результате, в каталоге проекта появляется файл «*-omegat.tmx» (либо «*-level2.tmx»; я склоняюсь к первому варианту в случае общей работы с OmegaT). Он содержит только те сегменты, которые были в исходных файлах. Именно этот файл и становится новой общей памятью переводов.

7. Далее получаем разностную память переводов между файлами, полученными на шагах 6 и 3. В этой памяти переводов содержатся переводы сегментов, которых нет в поддерживаемых версиях документации.
8. В следующем цикле перевода переводчики файл, полученный на шаге 6, кладут в подкаталог omegat, а файл, полученный на шаге 7, кладут в подкаталог tm.

Такая организация имеет как плюсы, так и минусы:
Плюсы: Сегменты, полученные на шаге 7, на данный момент не нужны, а они увеличивают вес памяти переводов. С другой стороны, в них может содержаться перевод, близкий к существующим и ещё не переведённым сегментам. Поэтому терять его тоже не хочется, возможно, он понадобится при переводе основной версии. Зато такая организация ускоряет автоматизированный перевод всех версий по окончании цикла. На скорости работы переводчиков она сказаться на должна, поскольку суммарный вес памяти переводов будет таким же.
Минусы: если кто-то из пользователей захочет осуществить перевод предыдущей и уже не поддерживаемой версии, то ему придётся сделать два прохода: первый – с памятью перевода в каталоге omegat с шага 6, второй – на уже переведенных файлах с памятью перевода в каталоге omegat с шага 7. В принципе, это не сложно и последовательность можно подробно описать. Да и мало кто будет это делать, я думаю. Либо можно просто перед переводом слить эти две памяти переводов. Либо можно даже хранить на FTP полную память переводов с шага 3.
Ещё одним плюсом предложенной последовательности является то, что когда мы перейдем на перевод новой версии первого уровня (5.x.x), вероятно, мы создадим новую память переводов project_save.tmx, зато в каталог tm можно будет кинуть файл с шага 6, не перегруженный лишними переводами.
В принципе, на данный момент мы работаем по шагам 1-5.

Структура каталогов и файлов на FTP
Нам нужны:
1. Исходная документация по каждой версии в архиве.
2. Переведённая документация по каждой версии в архиве.
3. Возможно, три памяти переводов - с шагов 3, 6 и 7 раздела “Память переводов”.
4. Заготовка каталога проекта со структурой подкаталогов и файлом проекта в архиве - для новых переводчиков. Глоссарий и память переводов я бы туда не клал, поскольку они постоянно меняются.
5. Глоссарий.

Тогда получается такая структура:

```
/qtrtt
   /qt4   проект перевода документации по Qt 4
     /doc-en    английские версии документации
       qt-doc-435-en.7z    qt-doc-[версия]-[язык].7z
       qt-doc-443-en.7z
       qt-doc-453-en.7z
       qt-doc-464-en.7z
       qt-doc-474-en.7z
       qt-doc-482-en.7z
     /doc-ru    русские версии документации
       qt-doc-435-ru.7z
       qt-doc-443-ru.7z
       qt-doc-453-ru.7z
       qt-doc-464-ru.7z
```

```
    qt-doc-474-ru.7z
    qt-doc-482-ru.7z
  /tm     все, связанное с памятью переводов
    /tm_2012.10     память переводов на дату слияния (год, месяц)
      iReset_diff_2012.07.06-2012.10.01.7z     разностная ПП пользователя
      project_save_qt4_2012.10.01.7z     ПП с шага 6
      project_save_qt4_full_2012.10.01.7z     ПП с шага 3
      tm_qt4_2012.10.01.7z     ПП с шага 7
    project_empty.7z     шаблон проекта с файлом проекта и структурой каталогов
    project_save_qt4_latest.7z     ссылка на последнюю ПП с шага 6
    project_save_qt4_full_latest.7z     ссылка на последнюю ПП с шага 3
    qt-glossary.utf8     глоссарий
    tm_qt4_latest.7z     ссылка на последнюю ПП с шага 7
/qt5   проект перевода документации по Qt 5 - в будущем
/qtcreator     проект перевода документации по Qt Creator
```