

Структурная кибер-лингвистика.

*Есть на всякий, есть на случай
В «Корабле» специалист –
Ваш великий и могучий
Структуральнейший лингвист.*

В XXII веке программирование - это прикладная наука, находящаяся на стыке лингвистики, психологии и инженерного дела, причем гораздо ближе к первым двум. Технологии шагнули далеко вперед, поэтому пусть вас не удивляет, что программирование XXII века сильно отличается от программирования XX века. Главное отличие заключается в том, что программирование оперирует не специальными машинными языками, а языком, чрезвычайно похожим на человеческий.

Основным элементом программирования XXII века являются "комплексные лингво-единицы" или "метаслова".

У киберлингвиста есть набор метаслов, которыми он может легко оперировать для составления программ. Этот набор называется обычно "метасловарем". Чем больше словарь - тем проще программировать.

Создание метаслова с нуля - долгий и очень трудоемкий процесс, требующий месяцы работы компьютеров. Однако киберлингвисты давно нашли способ его ускорить. Для этого они записывают метаобраз из человеческой памяти, а потом уже из него формируют метаслово по специальной формуле Светлова-Кусто. Этот способ быстрее в десятки раз, но требует дополнительного оборудования для записи метаобразов. К счастью, подобное оборудование есть на базе.

Метаслова, созданные с нуля, называются "синтезированными", а сформированные из метаобразов (то есть записанные из человеческой памяти) называются "естественными". Обычно киберлингвисты не придают особой разницы происхождению метаслова, однако стоит отметить, что у естественных метаслов в среднем заметнее мю-фактор, который большинство ученых склонны относить к помехам в программировании.

Программа - это предложение, составленное из метаслов. Это предложение составляется по правилам человеческого языка. Фактически метаязык позволяет человеку донести до кибера свои мысли с минимальным количеством ошибок - без дополнительного перевода на машинный язык, как это происходило в XX веке.

Теперь собственно модель.

Модель несложная, но требует внимательности и понимания. Вам понадобятся канцелярские принадлежности - бумага, ручка. Полезен будет степлер.

Метаслово описывается тремя параметрами:

- Уникальным кодом. Он состоит из одной русской буквы и трех цифр, например "Ю-001" или "К-264".
- Смысловой нагрузкой. Это слово русского языка в именительном падеже и пр. Например: "идти", "кибер", "белый".
- Мю-фактором. Мю-фактор записывается как число, перед ним ставится буква "мю" и обязательно указывается знак числа. Например: "мю +16", "мю =0", "мю -75".

Каждому киберлингвисту перед игрой выдают его "метасловарь", то есть список известных ему метаслов (у хорошего программиста их около десятка). По ходу игры вы можете дописывать в словарь новые слова - либо переписав их у другого киберлингвиста, либо записав новое слово (про это будет чуть ниже). Чтобы переписать метаслово из метасловаря в метасловарь - нужно добровольное согласие обоих программистов на эту операцию. Дальше просто берете и переписываете друг у друга те метаслова, которые пожелает. Метсловарь это знание, однако на игре вы можете моделировать его физический носитель листом бумаги с записью слов и указанием хозяина словаря.

Например, киберлингвист Вассинуарий получает от мастеров перед игрой такой список:

- "корова", К-211, мю(=0)
- "доить", Ч-007, мю(-14)
- "кибер", Ч-142, мю(+2)

Это довольно бедный набор, но с его помощью уже можно написать программу. Игрок выписывает слова на две бумажки А4, пишет вверху каждой из них "Метасловарь Вассинуария, не читайте, если вы не киберлингвист". Вассинуарий идет к другу Петру и переписывает из его словаря новое метаслово "кормить" (Р-212, мю+4).

Чтобы "написать" программу, игрок должен выписать на листе бумаги метапредложение и сопроводить его кодами метаслов и вычислением суммарного мю-фактора (см. пример). Этот лист вы отдаете тому киберу-игротехнику, на котором хотите протестировать программу. В процессе написания программы (метапредложения) вы можете: 1) добавлять туда любые предлоги и союзы, 2) изменять форму слов так, чтобы предложение лучше читалось по русски.

С точки зрения модели программа - это одно предложение на русском языке, которое описывает, что именно должен делать кибер. Особых формальных правил для формирования предложения нету, но оно должно быть понятным с точки зрения языка и общей логики.

"Кибер, пойдй в лес, набери дров" или

"Нужно двигаться по лесу до желтой отметки, собирая по пути лежащие на земле куски древесины, размером от 20 до 50 см" или

"Собери дрова, которые подойдут для топки печи, в лесу избегай монстров, если они нападают на тебя" - каждая из вышеперечисленных программ будет работать и поможет собрать дрова.

Пример: киберлингвист Вассинуарий пишет программу. Текст такой:

"Кибер доит и кормит корову"

Ниже он выписываем коды метаслов:

"Ч-142, Ч-007, Р-212, К-211"

И суммирует мю-факторы:

" +2+(-14)+4+0 = 2-14+4 = -8 "

Игротехник забирает у него лист с программой и уходит обрабатывать ее на мастерку. После обработки он начинает по игре производить ту деятельность, которую в него заложили. В нашем случае - пойдет, задаст корм корове и подоит ее. Он сделает это правильно и осмысленно - то есть положит корове правильную порцию еды, не перекормит ее. Он правильно подоит корову и поставит надоенное молоко в какую-то осмысленную емкость. Метаязык обеспечивает то, что он сделает все это правильно. Однако, так как Вассинуарий не написал в текст слова "поит", корова останется без питья. Отметим, что более умный программист использовал бы слова "ухаживает за коровой", и кибер бы и накормил, и напоил корову, и даже вывел бы ее пастись, если рядом есть подходящий лужок. Общее правило тут такое - текст программы трактуется в пользу игрока, пока она читается по-русски и не нарушает общечеловеческой логики.

Важно! Система обсчета программ будет полуавтоматической, поэтому не совершайте орфографических ошибок в кодах метаслов.

Влияние мю-фактора.

Как уже было сказано, в обычных программах мю-фактор действует как помехи. Если значение мю больше 30 или меньше -30, то программа будет работать неверно. Если значение мю больше 10, или меньше -10, программа может выполняться с ошибками. Идеальной считается программа с мю=0, хотя подобное качество обычно недостижимо, да и не требуется. Комбинирование разных слов поможет вам улучшить качество программы.

Моделирование записи новых слов

На игре у киберлингвистов не будет возможности создавать "синтезированные" метаслова - это слишком долго. Однако будет возможность записывать "естественные". Как следует из текста выше, для этого нужно записать метаобраз, а потом трансформировать его в метаслово по специальной формуле.

1. Для записи метаобраза киберлингвисту нужен другой человек - записывать метаобраз из собственной головы невозможно. Для простоты назовем этого человека "донором".
2. Важно! Использовать одного и того же донора чаще, чем раз в 12 часов не очень хорошо - последующие записи могут оказаться испорчены.
3. Также нужно особое оборудование "мнемограф". Он на базе один и сделать его предстоит сообща самим игрокам-кибернетикам. Этот прибор должен взаимодействовать с мозгом донора - шлемы, электроды на проводочках, закрепляемые на голове - думайте в этом направлении. Мы рассчитываем на вашу фантазию. Желательно, чтобы кибернетик и донор во время записи могли бы сидеть друг напротив друга, также донору нужно будет писать - озаботьтесь твердой поверхностью.
4. Вы подключаете донора к мнемографу и просите его описать что-то из его жизни, что вызывает у него сильные эмоции - неважно, какие. Это может быть воспоминание, моральные терзания или что-то подобное. Важно, чтобы оно вызывало эмоции. Писать надо недолго (2-3 минут) и немного (не больше страницы).
5. Далее возьмите написанный текст, тут же прочитайте его и обсудите его с донором. В процессе киберлингвист должен прояснить для себя (и донора), какую именно сильную эмоцию/эмоции тот испытывал. Желательно, чтобы в процессе обсуждения донор испытал их еще раз. Обсуждение не должно длиться дольше 5 или меньше 1 минуты.
6. Если есть желание продолжить беседу после окончания 5 минут, стоит отключить донора от мнемографа.
7. Сама беседа нужна тут для отыгрыша и прояснения точной эмоции. Для модели дальше важен лишь написанный донором текст. В мире игры он считается "метаобразом"- комплексной записью на каком-то твердом носителе информации. С этим текстом делается вторая часть записи.

Превращение "метаобраза" в "метаслово" производится так:

1. В верхнем левом углу нужно вписать имя и фамилию донора, а также время создания записи.
2. На листе с текстом киберлингвист выписывает сверху основную эмоцию, которую испытывал донор при записи этого метаобраза. Например, текст - это воспоминание о том, как донор в школе подрался с другим учеником, а Учитель отчитал его за это. В процессе беседы киберлингвист и донор понимают, что основной эмоцией связанной с этим воспоминанием является "стыд". Именно это слово программист выписывает на листе с метаобразом.
3. По таблице Светлова-Кусто выбирается часть речи в которую трансформируется метаобраз.

Часть речи	Эмоция	Примеры слов	Примечания
------------	--------	--------------	------------

существительное	радость, страх, отвращение		
прилагательное	удивление, печаль, вина		
глагол (или деепричастие)	гнев, любовь, стыд		
числительное			

1. Потом мы формируем одно конкретное слово, которое относится к этой части речи. Взять можно совершенно любое слово. Например: из эмоции стыд получили часть речи "глагол". Из него мы можем сделать любой глагол. Вообще любой: "бежать", "летать", "быть", "электрофицировать" и пр.
2. Все эти три шага выписываем на листе: "стыд = глагол = поить". После того, как выбранное слово записано на листе - менять его уже нельзя. Любые зачеркивания и помарки тут портят метаобраз - и его приходится записывать заного
3. Теперь в верхнем правом углу мы выписывает код метаслова. Выбрать можно любой - главное, чтобы этот код не совпадал бы ни с одним из ЛЮБЫХ ДРУГИХ кодов на игре. Если коды совпадут - все испортится. Советуем вам создать единый реестр кодов или придумать другую систему, чтобы они были бы уникальными.
4. Последний шаг - вычислить мю-фактор нового метаслова.
5. Для этого мы проходимся по тексту и вначале обводим все слова "я, мы, мое, мой, наше, наш" - то есть личные местоимения первого лица и притяжательные местоимения от них. В любых падежах. Каждое из таких слов увеличивает мю-фактор на единицу.
6. Потом также обходятся все личные местоимения третьего лица - "он, она, они, его, её, их". Каждое из таких слов УМЕНЬШАЕТ мю-фактор на единицу.
7. В итоге мю-фактор скорее всего будет положительным или отрицательным. Это нормально. Ошибка при вычислении мю-фактора - портит метаслово.
8. Не нужно просить игроков доноров употреблять или не употреблять в своем письме слова, влияющие на мю-фактор. Тут мы рассчитываем на вашу честность.

После того, как киберлингвист создал новое метаслово, он записывает его себе в метасловарь, а бумагу сдает киберу-игротехнику. Если это слово используется в программе - стоит сдавать их вместе (можно даже скрепить степлером). Игротехник проверяет, что метаслово было составлено верно. Если все в порядке - это метаслово потом можно использовать в программах.

О том, как используются программы

В распоряжении кибернетиков Филомены находятся около сотни КУРов (Кибер Универсальный Рабочий, модель 6) и два Кибера-Прототипа. КУРЫ на игре моделируются лишь виртуально, но их работа может быть нужна для других моделей (физика, энергетика, взаимодействие с лесом). Прототипы - это человекоподобные киберы, моделируются они игротехниками.

Прототипу игротехнику программу можно отдать в любой момент, если он не находится на подзарядке. Он её обработает и выполнит, если программа написана правильно.

Управление КУРами происходит так: раз в цикл ответственный программист с ЦПУ выбирает три (или меньше) задачи, которыми занимаются КУРЫ и определяет сколько процентов киберов будет ей заниматься.

Задачей всегда является одна "программа" - именно её загружают в память киберов. Допустим старший программист Федор ставит задачу КУРах на 2ой цикл. 40% будут заниматься починкой Периметра, а 60% пойдут в Лес, собирать образцы для биологов. По результатам этих заявок и качества написанных программ мастера выдадут результат работы киберстаи.

Совет: прежде чем запускать свежесозданную программу КУРах ее стоит хотя бы раз запустить на исполнение киборгу-прототипу.