rus.ie_nicola_tesla

Alexandr Kirilov (https://github.com/alexandrkirilov)

Информационный этюд. Тесла.

За основу данного этюда взято воспоминание Никола Тесла. Задача заключается в анализе ситуации описанной в этом воспоминании и создании модели информационного обмена по этой ситуации, пригодной для реализации на каком либо языке программирования.

Воспоминание Никола Тесла:

Возможности силы воли и самоконтроля чрезвычайно притягивали мое живое воображение, и я начал дисциплинировать себя. Если у меня было печенье или сочное яблоко, которое я до смерти хотел съесть, я отдавал его другому мальчику и испытывал Танталовы муки, с болью но и удовлетворением. Если передо мной стояла какая-то изнурительная задача, я набрасывался на нее снова и снова, пока не делал. Так я практиковался день за днем, с утра до ночи. Сначала это требовало сильного умственного усилия, направленного против склонностей и желаний, но шли годы, и это противоречие ослабевало, и наконец мои воля и желание стали одним и тем же. Таковы они и сегодня, и в этом лежит секрет всех моих успехов. Эти переживания настолько тесно связаны с моим открытием вращающегося магнитного поля, как будто они составили его существенную часть; если бы не она, я бы никогда не изобрел индукционный мотор.

Ключевая фраза для определения базы для описания ситуации: "Если передо мной стояла какая-то изнурительная задача, я набрасывался на нее снова и снова, пока не делал. Так я практиковался день за днем, с утра до ночи."

Есть некий человек который выполняет функцию решения какой-то задачи. Задачей является поиск соответствий шаблону в реальном мире, где искомый шаблон или группа шаблонов это то что было каким-то образом определено в мозгу человека (способы появления данных шаблонов лежат за пределами данной статьи). Воспринимаем этот искомый шаблон как данность.

Для понимания происходящего очень важно исключить наше собственное отношение к Николе Тесла как к гению и изобретателю. Он гений и изобретатель только с точки зрения результатов его действий. С точки зрения биологии - это обычный человек, мужчина, который как обычный человек обладает неким запасом энергии, он должен спать и есть для пополнения эти запасов, расходуемых на выполнение каких-либо задач. И он точно также должен ходить в туалет. Он точно также должен отдыхать в случае перегруженности (перегрева) средств выполнения функций. И как у каждого

обычного человека, любая тренировка (Значение слова "тренировка" в словаре Ожегова), с каждым повторением того или иного действия уменьшает энергозатраты и время на выполнения данного действия. У него есть штатный набор средств коммуникации с источниками информации: глаза, уши, и т.д.

В результате данного анализа мы получаем описание объекта (приложения), пусть он будет назван Nicola_Tesla у которого есть методы и параметры:

- var Energy_value: общая энергоемкость объекта, значение от 0 до бесконечности;
- method Restore_energy: функция суммирования текущего значения Energy_value с каким-то другим значением;
- method Restore_energy_by_sleep: функция наследующая Restore_energy с установкой значение по количеству энергии полученной во время сна;
- method Restore_energy_by_meal: функция наследующая Restore_energy с установкой значение по количеству энергии полученной во время приема пищи;
- method Check_energy: функция проверяющее значение переменной Energy_value и если она достигла какого-то значения выполнить какую-либо функцию привязанную к значению, некая абстрактная функция которая является основой для создания других функций;
- method Self_destruct: функция самоуничтожения объекта или выключения приложения (с точки зрения обыденной жизни выглядит ужасающе, но только потому что у нас срабатывает инстинкт самосохранения вызывающий страх как сигнал о достижении лимита для уничтожения, но с точки зрения информации и эффективного использования памяти это всего-навсего Garbage Collector Functionality, которое является средством выживания приложения или объекта);
- method Find_pattern_equality: циклическая функция которая выполняет поиск шаблонов в реальном мире, которая в конце каждого цикла уменьшает значение Energy_value на какое-то количество единиц в зависимости от используемых источников информации (например если используются только глаза, то это значение 25, если это глаза и уши это значение 45 и т.д.)
- и т.д.

При более детальном рассмотрении количество методов и переменных может быть разным. Для данной статьи, задачей которой является демонстрация принципов анализа этого достаточно.

Итак, в результате первичного анализа (первого витка спирали), мы получили некий объект (приложение) выполняющее функцию циклического поиска равенства искомому шаблону (pattern) по средствам опроса источников информации через средства коммуникации (глаза - зрение, уши - слух, нос - обоняние и т.д.), с возможностью комбинирования источников по средствам манипуляторов (рук), где в конце каждого

цикла происходит вычитание из текущего значения запаса энергии с проверкой необходимости смены состояния (state) объекта (приложения) для увеличения значения запаса энергии (для примера state_sleep: означает временно остановить выполнение цикла с сохранением состояния цикла на 8 часов, где каждая минута означит +1 к Energy_value).

С точки зрения локальных систем и приложений это идеальны условия. В реальном мире какому-либо объекту всегда будет что-то мешать.

Начнем с "Если у меня было печенье или сочное яблоко, которое я до смерти хотел съесть, я отдавал его другому мальчику и испытывал Танталовы муки, с болью но и удовлетворением."

Представим механизм "желаний", некий объект (приложение) встроенный в основной, выполняющий сохранение списка задач для различных механизмов, одним из которых является механизм циклического поиска который мы определили ранее. Для обеспечения защиты от перегрева других механизмов представим что у него есть некое значение которое при достижении 0 производит чистку списка желаний, а каждое новое желание добавляет значение тем самым отодвигая перезагрузку всего механизма желаний тем самым продлевая жизнь поставленных задач, в свою очередь вызывая риск "перегрева" всей системы. На основании этого мы получаем механизм который ставит желание "яблоко" которое добавляет, для примера +100 к жизни "механизма желаний" и самоудаляется после постановки и проверки того что значение увеличилось.

И т.д.

Отдельной темой является дисциплина и самоконтроль. Для понимания этого нужно представить Николу Тесла как некий сервер которой производит вычисления в сети с другими серверами, где дисциплина и самоконтроль могут быть firewall и user_policy.