Планы дальнейшей реализации

Разработка проекта наглядно показала, что развитие реагирования Beast это, по сути, развитие механизмов анализа накапливаемого опыта: от простейшего учета предыдущей реакции Оператора, до сложного анализа эпизодической памяти с группировкой, сортировкой, выделением значимого и т. п. Что означает: нет никаких интеллектуальных функций принятия решения кроме выборки из личного опыта. Поэтому дальнейшее развитие интеллекта Beast должно идти в том же направлении: какие варианты получения личного опыта можно организовать и что еще можно «выжать» из него?

### Творческая инициатива

Опыт можно получать, экспериментируя самостоятельно или наблюдая за экспериментами других. Поэтому нужен механизм анализа прошлого личного опыта, чтобы прогнозировать варианты своих экспериментов, а так же анализ сохраненного опыта Оператора, для отзеркаливания его прошлых действий.

### Общение с другими Beast

Общение с Оператором ограничивает Beast в объеме и скорости получаемых знаний. Намного эффективнее организовать обмен данными между разными Beast, чтобы они могли делиться полученным опытом, обучая друг друга. Если рефлекторный опыт после проверок на совместимость схем можно просто залить, то с автоматизмами так не получится: они индивидуально оцениваются. Такой опыт можно только «рассказать» - а как его «поймет» слушающий, зависит от его личного опыта, ассоциативной базы, текущего состояния. Возможно, придется изобретать язык общения – обще понимаемые вербальные символы. Для начала обратив внимание на механизмы коммуникаций насекомых и животных.

### Библиотечные модули

Чат-бот Beast взят в качестве примера реализации искусственной нервной системы по причине наглядности проявлений психических явлений и относительной простоты реализации. Но для дальнейшего развития систем проектирования искусственных живых существ желательно создать библиотечные модули с параметрами инициализации и развертывания в определенных условиях. Чтобы как в конструкторе можно было создавать самые разнообразные конфигурации на одних и тех же принципах и правилах, проверяя и уточняя их для специфических условий. Без стандартизации и унификации невозможно массовое распространение никаких идей.

### Инструмент разработки

У сожалению, язык Go несмотря на рекламируемые достоинства оказался во многом не удобным при реализации проекта. Пришлось вводить пульсацию, глобальные переменные, передаваемые между пакетами для избегания циклических ссылок и еще множество костылей, усложняющих и без того не простой для понимания проект. Пока рассматривается вариант перехода на Rust, так же под вопросом использование СУБД вместо файловой системы данных. Для прототипа с относительно небольшим объемом данных производительность терпимая, но если замахнуться на миллионы записей есть подозрение, что файловая система не выдержит. Тем более с примитивной системой «карт» от Goland, упрощенного аналога таблиц и запросов, сильно ограничивающая возможности работы с данными.