**Программная модель нервной системы живого существа**

В проекте реализован прототип нервной системы живого существа с потенциалом адаптивности, но с ограниченными каналами восприятия: вербальные сенсоры и пара десятков раздражителей, имитирующих внешнее воздействие. Существо названо Beast и по интерфейсу напоминает чат-бот, дополненный вспомогательными элементами регулирования параметров и настроек.

Наиболее общий признак, отличающий живое от не живого – наличие системы гомеостаза с жизненными параметрами с определенным оптимумом, выход за пределы которого для критически важных параметров приближает к противоположному состоянию – не живому. Если системе гомеостаза не удается стабилизировать такие отклонения, то существо становится не живым, мертвым.

Для параметров гомеостаза, которые напрямую не связанны с жизнеобеспечением, организована цикличность: при достижении параметром максимума, он держится 20 секунд и затем сбрасывается. Это позволяет создавать достаточно длительные периоды специфических контекстов реагирования.

Основной программный код реализован на Go, пульт управления на: html, php, js, ajax. Проект полностью открыт для ознакомления, исходники можно скачать на GitHub.

**Принципы построения адаптивной системы Beast**

Первая версия Beast, традиционно начавшись с попыток создания «нейросети с гомеостазом» быстро показала ненужность эмуляции функций нейронов, что позволило значительно снизить требования к ресурсной мощности компьютера. А то, что получилось в итоге, сильно отличается от привычного в ассоциации с ИИ понятия «нейросеть», это скорее блочное, модульное проектирование, где каждый новый уровень адаптации строится на основе предыдущего.

**Эволюционно обусловленное поэтапное развертывание системы**

Основа саморегуляции Beast – наличие у него встроенных жизненных параметров и системы гомеостаза для их стабилизации, которая в свою очередь задает первичную гомеостатическую цель реагирования Beast: стабилизировать вышедшие из нормы параметры. Система не дается в готовом виде, ее необходимо поэтапно настроить, проходя стадии развертывания:

1. **До рождения** – формируется система Гомеостаза, задаются начальные параметры, формируется база безусловных рефлексов.
2. **Рождение** – формируется база условных рефлексов, развивающая поведенческие навыки.
3. **Модификация** – на основе безусловных и условных рефлексов формируется база автоматизмов.
4. **Подражание** – добавляются скопированные действия Оператора как авторитарный вариант реагирования.
5. **Осмысление** – формирование у Beast модели понимания выполняемых действий, их значимости, собственного самоощущения, первые попытки поиска решений.
6. **Инициатива** – активация творческой инициативы, развитие методов поиска решений, провокации Оператора на нужные Beast действия, формирование доминант нерешенных проблем

Каждый этап строится на основе предыдущего. Нельзя перескакивать этапы и не желательно откатывать их обратно – это чревато каскадным обрушением накопленных поведенческих навыков с непредсказуемыми последствиями. Поэтапное развертывание системы имитирует ее эволюционное развитие, где очередной виток спирали это всегда продолжение предыдущего.

**Использование уникальных идентификаторов распознавания образа восприятия**

В природной реализации каждый нейрон имеет рецепторное поле, активность которого задает его специализацию, а выходной сигнал символизирует активность всего входного профиля. В свою очередь, активности рецепторного поля так же являются выходами предшествующего слоя, и в конечном итоге получаем при таком построении единственный сигнал, заменяющий собой активности множества предшествующих профилей. В проекте он используется как уникальный идентификатор образа и минимизировать условия возбуждения последующих структур.

**Древовидная структура распознавателя**

Каждый уникальный образ, воспринятый Beast, представлен в виде древовидной структуры с фиксированным числом узлов, задающим иерархию распознавания, где каждый узел означает все более усложняющийся тип распознавателя, имеющего более «примитивного» родителя. Например, образ рефлекторных действий это родитель для более сложного вербального образа, а вместе они образуют ветку совокупного интегрального образа. Подробнее об использовании образов будет ниже.

**Сопутствующая образу значимость**

Чтобы Beast мог реагировать осознанно, внешние раздражители должны иметь на него непосредственное воздействие. По аналогии с физической реальностью: сунул руку в огонь – ожог, больно – стало плохо – больше не буду туда совать руку. Тут, забегая вперед, показана цепочка осознания последствий реакции: действие – результат – оценка последствий – решение дальнейшего применения действия в таких ситуациях. Пока что отметим: если не будет имитации такого физического воздействия – не будет никакой осознанности потому, что действие будет обезличено. Нужна его **значимость**, сначала гомеостатическая, как непосредственно воздействует пусковой стимул (сунул руку в огонь – обжегся), затем при «мышлении» ассоциативная (если суну руку в огонь – обожгусь).

Значимость в проекте реализована через непосредственное воздействие не вербальных символов на параметры гомеостаза Beast. Это гомеостатическая значимость, которая в дальнейшем по мере развертывания системы дополняется ассоциативной: изменения некоторых параметров гомеостаза происходит теперь так же и при вербальном воздействии.

**Фиксация правил реагирования**

Если осознанное реагирование Beast начинается с оценки значимости пускового стимула и ответного действия, как он повлиял на гомеостаз, то разумная деятельность – с целеполагания, как спровоцировать Оператора на нужное в плане стабилизации гомеостаза действие. Но чтобы ставить цели, нужно уметь прогнозировать развитие ситуации, а для этого нужна эпизодическая память: знать, что уже было в похожей ситуации и предположить, что будет дальше. В проекте она реализована в виде структур Стимул – Ответ – Эффект, связанных в цепочку, фрагменты которой образуют правила реагирования. Они могут быть одиночными, в виде единственного кадра СОЭ, или групповыми, если последовательность кадров была выделена как многократно повторяемая. В любом случае конечное звено правила определяет конечный эффект – промежуточную цель, которых может быть несколько актуальных, но все они в конечном итоге завязаны на стабилизацию базовых параметров гомеостаза.

Работа с памятью, сортировка, группировка, поиск оптимальных цепочек достижений целей представляет собой процесс мышления Beast, сопровождаемый эмоциональными переживаниями: достижение цели – положительные эмоции, удаление от цели – отрицательные. Они могут временно перекрывать базовые гомеостатические по принципу: сделать шаг назад, чтобы потом прыгнуть на 2 шага вперед. А могут теоретически и полностью подмять под себя какую то часть гомеостаза: чем выше и важнее промежуточная цель, тем она менее достижима, и потому постоянно доминирующая.

**Доминанта не решенной проблемы**

Если цель не достигнута по каким то причинам, но она важна – возникает доминанта не решенной проблемы, которая периодически активируется при подходящих условиях: нет опасной ситуации, возникает окно для новых попыток решения. Соответственно нужны механизмы, определяющие такое окно, оценивающие очередной прогноз решения и записывающие его в виде ментальных правил (логических цепочек).

**Система гомеостаза Beast**

При запуске программы запускается генератор пульсации с частотой 1 сек, который синхронизирует выполнение следующих процедур:

1. Коррекция текущих значений базовых потребности и базовых контекстов при внешних воздействиях.
2. Определение текущего состояния по каждой базовой потребности, затем общего интегрального состояния.
3. Взаимодействие с Пультом связи.

Основа гомеостаза – **Базовые потребности (параметры)**, количество которых определяет сложность системы и возможную вариабельность адаптации. Базовые потребности задают первичную гомеостатическую цель адаптации – стабилизацию вышедших из нормы базовых потребностей. Они могут быть нарастающие и убывающие при пульсации, что определяет тип стабилизации: повышение или понижение значения параметра, если он соответственно станет выше или ниже критического значения.

Критическое значение базовой потребности определяет **Порог (компаратор)**, пересечение которого означает переход из одного **Базового состояния** в другое. Оно может принимать три значения, в зависимости от текущих значений базовой потребности и порога:

* **Плохо** – если значение лежит в области после Порога у нарастающей базовой потребности, или в области до Порога для убывающей.
* **Норма** – если значение лежит в области до Порога у нарастающей базовой потребности, или в области после Порога для убывающей.
* **Хорошо** – состояние возникающее при переходе из Плохо в Норма. Оно всегда кратковременно, в данной версии Beast длится 50 пульсов.

|  |  |
| --- | --- |
| bp1_narastanie.png | bp1_ubiwanie.png |

На графиках красным показаны критические зоны с нарастающей величиной ухудшения (значимости) состояния по мере удаления от Порога: чем дальше от порога, тем хуже по экспоненте. В данном случае в градации от 0 до 10. Показаны варианты с положительными и отрицательными величинами значимостями, что не принципиально.

Если текущее Базовое состояние Норма, то диапазон значений от текущего до Порога определяет Зону комфортного состояния, если текущее Базовое состояние Плохо, тогда диапазон значений от него до Порога определяет Критическую зону. Отсюда определяются уровни стабилизации Базовых потребностей:

* **стало Лучше** – если произошло приближение или заход в Зону комфорта
* **стало Хуже** – если произошло удаление или выход из Зоны комфорта

В общем смысле стабилизация Базовой потребности означает нахождение в Зоне комфорта, а выход из зоны – необходимость принятия мер для возврата в нее.

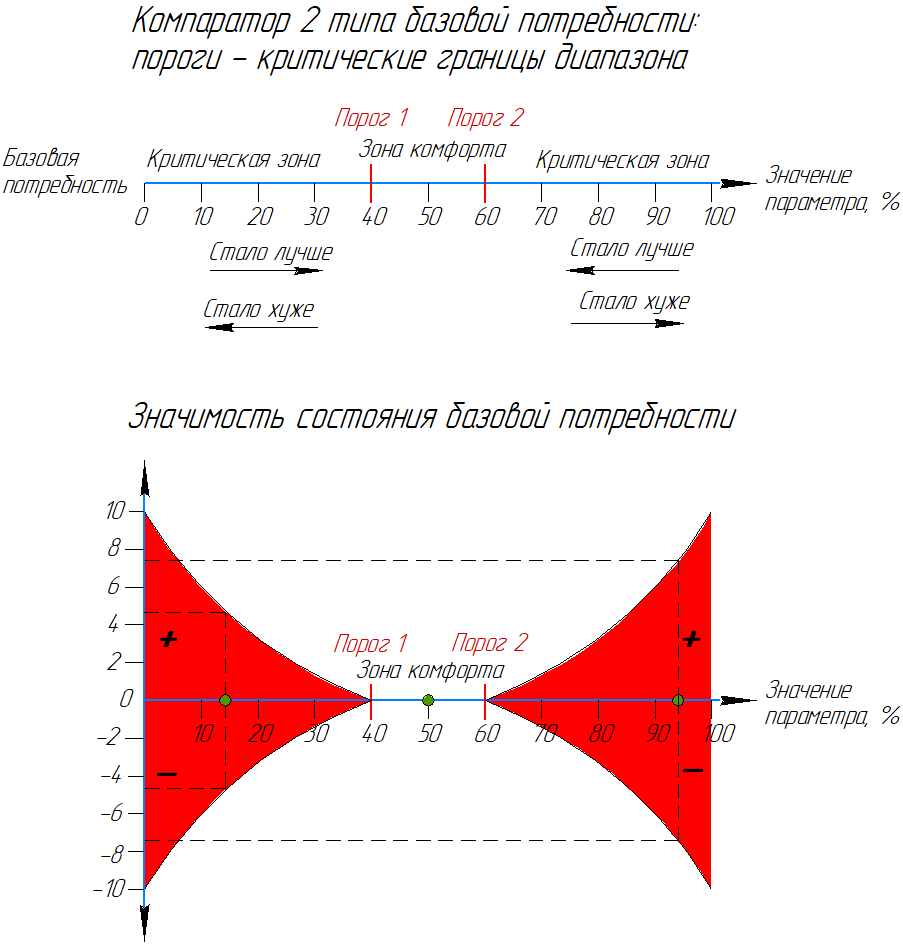
При проектировании Beast возник теоретический вопрос – возможна ли самоадаптирующаяся система с одной базовой потребностью? В принципе да, можно даже уверенно сказать на базе какой: энергобаланс. Всем нужна энергия, кто не следит за ней – тот умрет. Но если вся адаптация будет крутиться только вокруг подзарядки, то например робот-пылесос быстро сообразит, что лучший способ сохранить заряд – не слезать с базы. Это станет его единственной целью и смыслом жизни, все прочее, включая заложенные разработчиком алгоритмы уборки, станут вторичны. Чтобы заставить его шевелиться придется прошить как минимум еще одну потребность, противоречащую первой – страсть к чистоте. И начнется борьба противоположностей с поиском компромиссов: начнешь убирать – будет потеря энергии, не будешь убирать – будет потеря жизненных ориентиров, тоска и уныние. И чем больше базовых потребностей, тем сложнее и интереснее жить. Это можно сравнить с жонглером, крутящим тарелки на шесте: можно наловчиться с одной тарелкой подобрать такой ритм, чтобы она ровно и стабильно крутилась, но уже с парой тарелок такой номер не пройдет: пока одну стабилизируешь, другая идет вразнос. Придется метаться между ними в поисках оптимума, который тем сложнее найти, чем больше тарелок.

Другой вопрос, который возникает, глядя на систему гомеостаза Beast: почему взято именно три базовых состояния Плохо/Норма/Хорошо? Тут есть нюанс: если мы говорим о рефлексах, как неизменных предопределенных действиях при распознавании пускового образа, то в принципе достаточно 2 состояний: Плохо, запускающее стабилизирующее рефлекторное действие, если Базовая потребность вышла за Порог, и Норма, если оно вернулось в зону комфорта и надо остановить рефлекс. То есть просто Вкл/Выкл. Состояние «Хорошо» тут не нужно потому, что рефлекс не модифицируем в принципе и будет выполняться в любом случае: муха тупо бьется о стекло пока ее случайно не сдует ветром или она не обессилит. «Хорошо» могло бы пригодиться для ситуации **закрепления успешной реакции**, но рефлексы это явно не тот случай. В проекте Beast модифицируемые реакции названы автоматизмами, о них будет ниже. И тем не менее базовое состояние «Хорошо» присутствует уже на уровне рефлексов для… закрепления успешной реакции. Здесь нужно уточнить, введя следующее пояснение разницы между рефлексами и автоматизмами:

*Адаптация на уровне безусловных рефлексов формируется в течении жизни многих поколений путем естественного эволюционного отбора случайных мутаций. Адаптация на уровне автоматизмов формируется в течении жизни одной особи механизмами нервной системы. При этом в обоих случаях, для закрепления успешно найденного действия нужно состояние «Хорошо».*

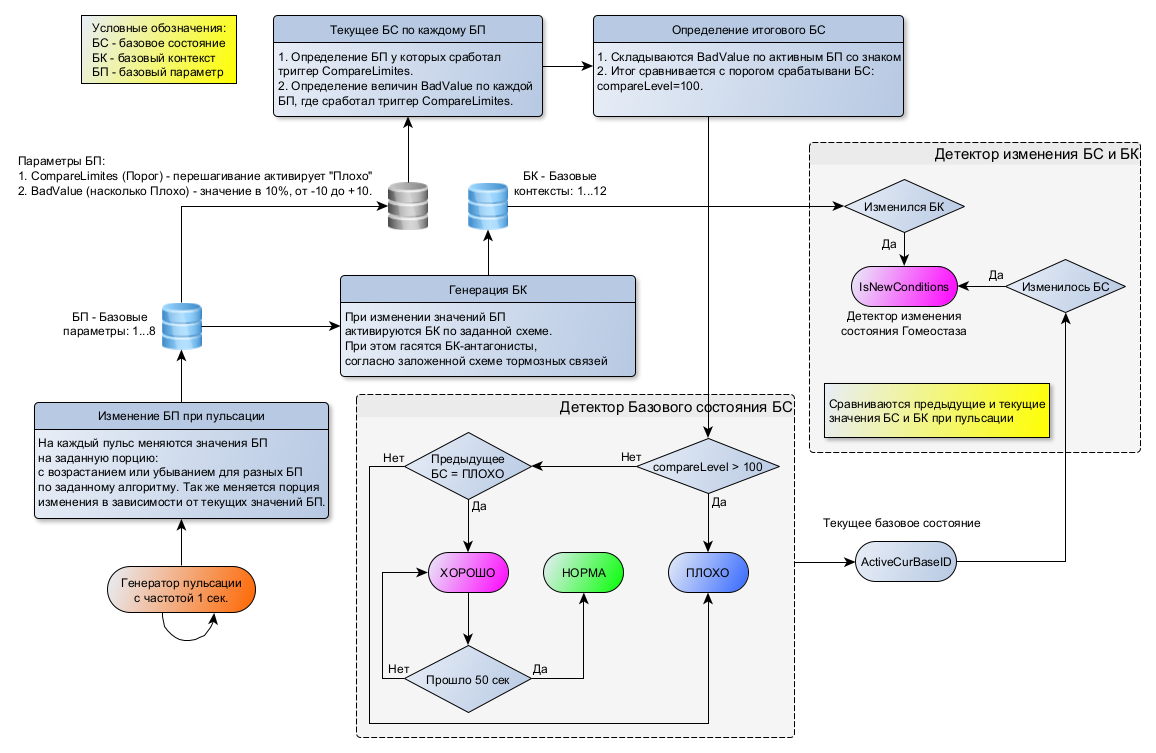
Поэтому, уже начиная с рефлекторного уровня реагирования, есть механизм закрепления успешной реакции через функциональность состояния «Хорошо», к которому привязаны соответствующие рефлексы, подкрепляющие только что успешно выполненный рефлекс, усиливая его положительный эффект.

В текущем проекте Beast используются Базовые потребности с одним порогом, но в принципе возможны схемы с несколькими порогами. Ниже показан вариант с двумя.



Два Порога в этом случае задают границы зоны комфорта, и можно сказать, что схема с одним Порогом является частным случаем общей схемы с несколькими. Просто в этом случае в качестве Порога используется граничные значения Базовой потребности: 0 или 100. В зависимости от типа потребности – нарастающей или убывающей.

На основе локальных базовых состояний определяется **Общее интегральное базовое состояние**, которое является первым уровнем интегрального пускового образа рефлексов и автоматизмов. Для его получения учитываются уровни Плохо (насколько удалено текущее значение от Порога) по каждой потребности в виде суммарного отклонения, которое сравнивается с общим Пороговым значением 100, при превышении которого активируется интегральное Базовое состояние Плохо. Соответственно, если суммарное по всем Базовым потребностям Плохо меньше Порога – активно Базовое состояние Норма, а при переходе из Плохо в Норма активируется Хорошо.



**Безусловные рефлексы**

вапвап