**0 стадия**

я

**3 стадия**

На предыдущих стадиях Базовое состояние служило одной из частей Пускового образа, запускающего Рефлексы. Но такое реагирование никак не учитывает возможную неадекватность Рефлекса при изменившихся внешних условиях, что означает единственную возможность адаптации только при смене поколений и удачной новой мутации, формирующей другой, более подходящий Рефлекс. Для адаптации в течение жизни одной особи нужен дополнительный маркер в виде оценки изменения Базового состояния: стало лучше или стало хуже после выполнения ответных действий. Но так как рефлекторное реагирование, не подразумевающее никаких оценок, и выполняемое в любом случае при активации Пускового образа, уже не годится, то нужен принципиально новый тип реактора – **Моторный автоматизм**.

**Моторный автоматизм** – ответная реакция, способная модифицироваться (блокироваться, изменяться) в зависимости от оценки изменения текущего Базового состояния Плохо/Норма/Хорошо по принципу: стало лучше/стало хуже. При улучшении Базового состояния автоматизм закрепляется, при ухудшении блокируется или модифицируется. Автоматизмы в отличие от Рефлексов, формируемых эволюционным отбором полезных мутаций, позволяют особи адаптироваться в течение ее жизни. Автоматизм имеет приоритет активации над Рефлексом, что означает: Рефлекс активируется только если нет автоматизма на аналогичный пусковой образ.

Запущенный Моторный автоматизм активирует интегрированный Образ ответных действий ActionsImageID, включающий в себя действие, фразу, тон сообщения и настроение, с регулируемой силой Energy в градации от 1 до 10, при этом только что созданный автоматизм выполняется с усилием по умолчанию 5. Это позволяет подстраивать автоматизм для текущих условий, меняя образ и силу воздействия. Успешность выполнения определяется через параметр Usefulness, показывающий (бес)полезность Моторного автоматизма.

Подробнее структура Моторного автоматизма:

* **BranchID** – код привязки автоматизма, в качестве которого может быть:
  + код дерева автоматизмов
  + код образа действий оператора с пульта ActivityID
  + код фразы оператора с пульта VerbalID
* **Usefulness** – (бес)полезность автоматизма: от -10 вред, через нейтральный 0 до +10 польза
* **ActionsImageID** – образ ответных действий автоматизма
* **Energy** – сила автоматизма, от 1 до 10, по умолчанию = 5
* **Belief** – степень уверенности выполнения автоматизма. Принимает значения:
  + **0** – предположение
  + **1** – чужие сведения,
  + **2** – проверенное собственное знание

Уверенность Belief в авторитарном автоматизме высока в период авторитарного обучения и сильно падает в период собственной инициативы, когда нужно на себе проверить, а даст ли такой автоматизм в самом деле обещанное улучшение. Только один из автоматизмов, прикрепленных к ветке, может иметь Belief = 2, проверенное собственное знание. Если задается Belief = 2, остальные Belief = 2 становится Belief = 0.

* **Count** – надежность. Число использований с подтверждением (бес)полезности Usefulness. В случае, если в результате автоматизма его Usefulness изменит знак, то Count обнулится, а при таком же знаке - увеличивается на 1.
* **GomeoIdSuccesArr** – массив кодов гомео-параметров, которые улучшает автоматизм.

**ActionsImageID** – интегрированные образы действий оператора или Beast. Фактически структура повторяет ActionsImage из рефлексов и позволяет сохранять как образы действий в автоматизмах, так и образы действий оператора, отражаемые в дереве моторных автоматизмов.

Структура ActionsImageID**:**

* **ActID** – массив кодов действий
* **PhraseID** – массив кодов фраз
* **ToneID** – тон сообщения оператора
* **MoodID** – настроение оператора

Моторные автоматизмы обычно привязаны к ветке Дерева моторных автоматизмов, которое задает иерархию активации автоматизма от простых одиночных стимулов, до сложных составных, когда для его активации учитываются несколько составляющих. То есть дерево формирует интегрированный пусковой стимул, который задает не просто комбинации пусковых стимулов, но и правила таких комбинаций согласно иерархии уровней: можно комбинировать уровни 1,2 – 1,2,3 – 1,2,3,4, но нельзя: 1,3 – 2,4 – пропуская уровни.

Исключение сделано для комбинации стимулов «действие» и «фраза» с пульта. В этом случае Моторные автоматизмы могут быть привязаны к отдельным значениям интегрированного пускового стимула: только к фразе или только к действиям с пульта. То есть в этом случае автоматизм активируется по фразе или кнопке с пульта без учета тона и настроения.

Дерево автоматизмов активируется при следующих условиях:

1. При любых событиях с Пульта – так же как дерево Рефлексов, но если к ветке привязан автоматизм, то он выполняется преимущественно, блокируя Рефлексы потому, что уже было произвольностью преодолено действие Рефлекса при выработке автоматизма. Такой автоматизм обладает меткой успешности Usefulness > 0. Успешность 0 означает предположительный вариант действий, а успешность < 0 заблокированный вариант действия. Поэтому к ветке может быть прикреплено множество неудачных и предположительных автоматизмов и только один удачный. Более удачный результат переводит ранее удачный автоматизм в предполагаемые.
2. При произвольной активации отдельных условий. Отсутствие подходящей для данных условий ветки дерева вызывает Ориентировочный Рефлекс привлечения внимания к активной ветке с осмыслением ситуации и рассмотрением альтернатив действиям (4 уровня глубины рассмотрения). При формировании нового предположительного действия создается новая ветка дерева и к ней прикрепляется автоматизм. Т.е. сами по себе новые условия не создают новой ветки, а только новый автоматизм, а пока нет автоматизма, будет Ориентировочный рефлекс.

У дерева Моторных автоматизмов 6 фиксированных уровней, задающих иерархию активации Автоматизма:

1. Основание дерева.
2. Базовое состояния: Плохо/Норма/Хорошо
3. Эмоция, может произвольно меняться, независимо от базовых контекстов, например при базовом состоянии Плохо может быть позитивная эмоция.
4. Образ сочетания действия с Пульта
5. Образ контекста сообщения: сочетание тона и настроения. Например: "Обычный, Хорошее"
6. Первый символ фразы для оптимизации поиска по дереву
7. Фраза

До 6-го уровня такая структура это полный аналог Условным рефлексам, только вместо сочетаний контекстов используется эмоция.

База Моторных автоматизмов формируется 2 способами:

1. Клонированием существующих безусловных и Условных рефлексов. В этом случае просто создаются Моторные автоматизмы в виде копий Рефлексов: копируются их пусковые образы и ответные действия. Таким образом сохраняется иерархия реагирования и появляется возможность модификации наработанного эволюционным отбором опыта. Ведь Рефлексы нельзя модифицировать, и в то же время для адаптации в течении жизни одной особи это как то нужно сделать. Решением будет модифицировать «копии Рефлексов».
2. Отзеркаливанием действий оператора, которые он совершил в ответ на действия Beast, наработка примитивного личного опыта Beast.

В третьей стадии в процессе диалога Beast с оператором запоминается предыдущий ответ оператора как пусковой образ и к нему привязывается ответное действие Beast. В результате получается, что Beast использует ответы оператора в виде авторитарного указания, как надо реагировать. Однако, таким образом можно только уточнять уже существующие автоматизмы. Для случая их отсутствия выполняется следующий трюк: Beast просто повторяет пусковой образ оператора, провоцируя его тем самым дать свой ответ на него, который и запишется как вариант реагирования в виде **Зеркального автоматизма**.

**Зеркальный автоматизм** – моторный автоматизм, сформированный в процессе подражания действиям оператора (импринтинга) с изначально установленной высокой степенью доверия (авторитарно).

Пример такого диалога с отзеркаливанием:

**Оператор:** привет, как дела?

**Beast:** привет, как дела? (нет автоматизма на пусковой образ «привет, как дела?», активируется его повтор)

**Оператор:** нормально (формируется зеркальный автоматизм: привет, как дела? - нормально)

**Beast:** нормально (нет автоматизма на пусковой образ «нормально», активируется его повтор)

В результате при следующем диалоге с такими же пусковыми стимулами Beast активирует сформированные зеркальные автоматизмы и будет отвечать, подражая оператору:

**Оператор:** привет, как дела?

**Beast:** нормально

Таким образом происходит передача опыта реагирования от оператора к Beast путем прямого подражания, как это делает попугай. Однако очевидна и ограниченность такого подхода: шаги диалога не связаны между собой общим контекстом, что приводит к неадекватным реакциям Beast. Он периодически отвечает невпопад потому, что не учитывает контекст диалога, который он еще не умеет определять. Поэтому следующим этапом развития реагирования будет прогнозирование, которое в свою очередь так же разделятся на уровни:

1. прогноз последствий своих и операторских действий, что позволит более точно выбирать подходящий автоматизм.
2. обобщение своих и операторских действий, что позволит определять контекст поведения

**4 стадия**

При создании зеркальных автоматизмов запоминался только один предыдущий шаг диалога, но для возможности предсказания последующих шагов нужно запоминать цепочки диалога, чтобы пройдя до конца найденной в памяти по начальным шагам текущего диалога цепочки, посмотреть ее предположительный результат. Такое поведение Beast означает активацию у него новой структуры – эпизодической памяти, которая может быть 2 типов: объективная, активированная действиями оператора, или субъективная, активированная внутренними действиями Beast.

Эпизодическая память представляет собой цепочку последовательных эпизодов событий, в которых фиксируется:

* **Trigger** – образ действий оператора с пульта или текущей ситуации
* **Action** – образ ответных действий Beast ActionsImage
* **Effect** – эффект от действий (-1 плохой, 0 нейтральный, +1 хороший)
* **LifeTime** – текущее время в пульсах от начала активации Beast на момент формирования звена
* **Type** – тип образа:
  + **0** – объективный образ (внешний Ориентировочный рефлекс)
  + **1** – субъективный (произвольный Ориентировочный рефлекс)

Каждый эпизод представляет собой зафиксированную по времени через LifeTime тройку одного типа *Стимул – Ответ – Эффект*, где:

* **Стимул (Trigger)** – активация Дерева автоматизмов по действиям с Пульта оператора или по изменению гомеостатического состояния.
* **Ответ (Action)** – сформированные действия Beast на данный Стимул. Ответ всегда бывает только в виде образа действий ActionsImage.
* **Эффект (Effect)** – результат Ответа: -1 негативный, 0 без изменения, +1 позитивный.

Эта тройка сохраняется в TriggerAndAction и далее будет называться одиночным **Правилом** Rules. Цепочка последовательных кадров-Правил одного типа Type представляет собой **Групповое правило**.

В эпизодической памяти фиксируются моменты активации дерева **Ментальных автоматизмов**, которые происходят по следующим причинам:

1. Не нашлось штатного автоматизма для текущего пускового образа
2. Штатный автоматизм оказался заблокированным из за прошлых неудачных активаций
3. В течении 20 пульсов не было ответа оператора, нет пускового образа для реагирования

При фиксации пишутся эпизоды в хронологическом порядке цепочки, случившиеся в течении последних 180 пульсов. Это значит, что нужна еще промежуточная, короткая память, размером в 180 пульсов, аналогичная по структуре с эпизодической, которая постоянно перезаписывается. А эпизодическая память хранит только фрагменты событий по факту прерывания, между которыми могут быть значительные пробелы, когда штатные автоматизмы успешно активировались и не требовалось прерываний для осмысления. По этой причине по мере увеличения базы автоматизмов реагирование становится все более «моторным» и субъективное ощущение времени ускоряется. А в начальный период наоборот, прерывания на осмысление непонятных ситуаций возникают постоянно, и субъективное время «растягивается».

**Ментальный автоматизм** – специфический автоматизм, активирующий Информационные функции поиска решений текущих задач, например поиск наилучшего правила реагирования Rules, в зависимости от текущего состояния Информационной среды Beast.

**Информационная функция** – разные методы получения информации, систематизации, поиска и т.п. с целью найти верный Метальный образ действия MentalActionsImages моторного автоматизма, а если нет, то создания нового ментального автоматизма для продолжения итерации поиска.

**Ментальный образ действий** – интегрированный образ действий, способный менять Информационную среду.

**Информационная среда Beast** – интегративная информационная среда условий текущего состояния Beast, является основой самоощущения Beast, отражая изменения Базового состояния, Активных Базовых контекстов, Активных действий с пульта только при Ориентировочном рефлексе и осмыслении результатов.

Структура информационной среды:

* **IsSleep** – признак сна Beast. True – спит, False – бодрствует.
* **PsyEmotionImg** – текущая эмоция, может быть произвольно изменена
* **Danger** – признак опасного внутреннего состояния. True – опасное, False – не опасное.
* **Mood** – общая градуированная оценка гомео-настроения в значениях от -10 Плохо через 0 до +10 Хорошо
* **PsyMood** – субъективно ощущаемая оценка, текущее осознаваемое настроение, которое можно произвольно изменять. Задается значениями:
  + **-1** – Плохое настроение
  + **0** – Нормальное настроение
  + **+1** – Хорошее настроение
* **PsyActionImg** – текущий образ сочетания действий с Пульта Activity. Хранится до следующего любого действия или фразы с Пульта но не более 100 пульсов
* **PsyActionImgPulsCount** – текущее время обновления в пульсах
* **PsyVerbImg** – текущий образ фразы с Пульта
* **veryActualSituation** – признак опасной ситуации, необходимость срочных действий. True – опасная ситуация, False – не опасная.
* **curTargetArrID** – массив кодов параметров гомеостаза как цели для улучшения в данных условиях

Ментальный автоматизм, по аналогии с Моторным автоматизмом, привязан к дереву ментальных автоматизмов, имеющего 4 фиксированных уровня, не считая базового нулевого. Это **Дерево понимания**, в конечных узлах которого накапливаются Ментальные автоматизмы.

Структура Дерева понимания:

* **Mood** – ощущаемое настроение:
  + **-1** – Плохое
  + **0** – Нормальное
  + **+1** – Хорошее
* **EmotionID** – эмоция, может произвольно меняться
* **SituationID** – код объекта структуры понимания SituationImage, может произвольно меняться.

SituationID определяет основной контекст ситуации, определяемый при вызове активации дерева понимания. Если этот контекст не задан то сначала он выбирается наугад (для первого приближения) из более важных существующих, но потом дерево понимания может переактивироваться с произвольным заданием контекста. От этого параметра зависит, в каком направлении пойдет информационный поиск решений, если не будет запущен штатный автоматизм ветки (ориентировочные реакции). Более частный, целевой контекст ситуации определяется следующим параметром PurposeID.

* **PurposeID** – код образа желаемой при данных условиях цели.

PurposeID сначала наследует PurposeGenetic, но может произвольно меняться, в том числе после подсказки оператором в результате осмысления ответа оператора и запуска Ментального автоматизма корректировки цели с перезапуском дерева понимания. Для достижения этой общей цели в цепочках Ментальных автоматизмов определяются промежуточные цели так, что каждый Ментальный автоматизм оценивается успешным при появлении состояния, соответствующему данной промежуточной цели, а конечное звено цепи должно стремиться к соответствию PurposeID.

Начиная с 4 стадии реагирование отличается от предыдущих стадий тем, что теперь после выполнения действий оператора происходит процесс его осмысления через функцию **Consciousness** для выбора оптимального автоматизма, который может отличаться от того, что привязан к активировавшемуся узлу Дерева автоматизмов и должен запуститься «по умолчанию».

**Consciousness** – главная, постоянно активная с каждым ориентировочным рефлексом функция циклов осмысления для поддержания информационной среды и произвольности. Изолированная от непосредственных воздействий и поэтому самостоятельная система оценки и корректировки состояния, происходит с формированием опыта произвольности выполнения Правил и их выбора для данных условий, в частности, Правил относительно самого себя (самосознание).

Функция осмысления Consciousness имеет четыре уровня:

1. есть ли штатный Моторный автоматизм и нужно ли его менять
2. попытка использования примитивных Правил
3. попытка найти решение, используя всю текущую информацию с учетом срочности
4. доминанта нерешенной проблемы, только если нет срочности (Творчество)

**Объективная активация** – активация Ориентировочным рефлексом регистрации новой ситуации, ОР новизны

**Субъективная активация** – активация «внутренним» (произвольным) Ориентировочным рефлексом, ОР произвольности выбора

Функция Consciousness вызывается как объективным Стимулом, так и субъективным, т.е. по инициативе в самой Consciousness (при срабатывании ментальных автоматизмов). В случае субъективного вызова (только после объективного) возникает цикл рекурсии, по результату которого возвращается итог выполнения функции. Результат рекурсии – всегда запуск Моторного автоматизма (при запуске Ментального автоматизма с таким действием) или переактивация системы.

Если не сработал первый уровень рекурсии, делается попытка найти последний фрагмент последовательности Правил в Эпизодической памяти, в которой подряд пишутся все активации consciousness в виде Правил.

Сonsciousness названа осмыслением потому, что в ней реально с каждым вызовом происходит обновление информационной картины текущего состояния (о котором говорил Дж. Тонони), что является контекстом осмысления. Объективный вызов происходит по внешнему ориентировочному рефлексу, а вызов ментальный можно назвать внутренним ориентировочным рефлексом. При вызове так же пишется эпизодическая память и еще Кратковременная память shortTermMemory.

При каждом осмыслении становится доступной новая информация, получаемая с помощью Ментальных автоматизмов, но сам процесс ее получения оказывается субъективно недоступен (нет записи промежуточных данных).

При срабатывании Ментальных автоматизмов перезапускается Consciousness для того, чтобы организовать цикл обдумывания с помощью ментальных автоматизмов, который можно прервать следующей объективной активацией Consciousness. То есть Consciousness может активироваться субъективно множество раз между двумя объективными активациями. Таким образом осмысление происходит очень быстро, но если прервалось, то состояние на момент прерывания остается доступным для следующего осмысления. Поэтому при объективной активации сначала нужно посмотреть, есть ли прерванные цепочки осмысления и, если есть, выбрать, с какой продолжить осмысление.

В основе организации осмысления лежат цепочки последовательных вызовов Ментальных автоматизмов, которые всегда начинаются с одного из узлов активной ветки Дерева понимания, что фиксируется в массиве goNextFromUnderstandingNodeIDArr (поиск по узлу дерева понимания).

**goNextFromUnderstandingNodeIDArr** – представляет собой память о том, как шли размышления до конечного звена: создания успешного Моторного автоматизма. При этом ранее созданные успешные автоматизмы могут стать неуспешными в каких-то условиях и построение цепи продолжится. Но последовательности Правил в эпизодической памяти хранят случаи успешного действия и они могут использоваться.

От одного узла Дерева понимания может начать расти множество цепочек в зависимости от активного узла Дерева автоматизмов. Такие цепочки НЕ могут ветвиться на каждом из звеньев т.к. всегда продолжаются для текущей активности дерева автоматизмов: если сменится активная ветка, то станет активной и другая цепочка. Но все оказывается связанным (обобщенным) через Кратковременную память последовательности запусков звеньев всех цепочек.

Цепочки организуются с помощью структуры ветвителя goNext:

* **MotorBranchID** – узел AutomatizmNode
* **FromID** – код родительского goNext
* **NextID** – код дочернего goNext
* **AutomatizmID** – код Ментального автоматизма

Каждое звено типа goNext позволяет направлять поиск к следующему или предыдущему звену, обеспечивая целостность и направление развития цепочки. Фактически так определяются последовательности Ментальных автоматизмов цепочки размышления.

**Уровни осмысления Consciousness**

**Уровень 2 - попытка использования примитивных Правил**

Функция getRulesFromEpisodicsSlice позволяет найти последнее известное Правило по цепочке последних кадров Эпизодической памяти (шаблон решений) с учетом шаблона уже реализованных Правил. Последовательность шаблона использует прежний опыт цепочек Стимул – Ответ – Эффект с ожиданием очередного Стимула для последнего нахождения подходящего Ответа. Каждый раз, находя последнее правило в данной ситуации, оно может использоваться для того, чтобы перейти к следующему известному правилу или, если такого нет, начать формировать новое решение Ментальным автоматизмом получения информации с запуском ментального осмысления. Получается ветвление дерева решений (основанного на эпизодической памяти) по каждому Стимулу. Такой поиск одинаково работает как для кадров объективной Эпизодической памяти, давая решение для запуска Моторных автоматизмов, так и для кадров субъективных, давая решения запуска Ментальным автоматизмам.

**Уровень 3 - попытка найти решение, используя всю текущую информацию с учетом срочности**

На третьем уровне Consciousness происходит сложное использование Правил с учетом всей текущей информации окружения. Причем здесь используются не только Правила, полученные объективно, но и полученные субъективно, что оптимизирует процесс мышления. При этом каждый ментальный автоматизм определяется действием в виде Ментального образа действий MentalActionsImages.

Структура Ментального образа действий:

* **activateBaseID** – активация эмоции
* **activateEmotion** – активация эмоции
* **activateInfoFunc** – вызов Информационной функции
* **activateMotorID** – запуск Моторного автоматизма по результатам Информационной функции создания автоматизма

Осмысление на третьем уровне вызова Consciousness происходит с помощью цепочек ветвления goNextFromIDArr, начинающихся от узла дерева понимания goNextFromUnderstandingNodeIDArr. Осмысление – это активация последнего звена цепочки и выбор: если там есть уже хороший Моторный автоматизм, то просто запустить его. Хотя испытанный хороший автоматизм и сам вызовется, т.к. запишется штатным для данного узла дерева автоматизмов в Периоде ожидания. Если нет, то начинать формировать следующий Ментальный автоматизм и помещать его в звено продолжения цепочки. И как уже говорилось, запуск Ментального автоматизма сопровождается обязательным перезапуском Consciousness.

Целью процесса осмысления на третьем уровне является нахождение наиболее подходящего к текущим условиям действия моторного автоматизма – Ответа. Тогда он запускается с периодом ожидания. По результату Периода ожидания записывается ментальное Правило из всего цикла осмысления. Если хорошо, то данная цепочка так и заканчивается запуском Моторного автоматизма, если плохо – формируется следующая цепочка с выбором другого MentalActionsImages и т. д. То есть формирование следующего звена цепочки идет с учетом опыта предыдущих.

Поиск MentalActionsImages для следующего узла цепочки ветвления идет по ментальным Правилам или по функции случайного или прописанного в редакторе подбора с учетом опыта предыдущих (Переактивация Базового состояния или эмоций выбираются на 5-й ступени развития). Структура ментального Правила MentalTriggerAndAction начинается или с ментального действия MentalActionsImage или с моторного ActionsImage, потом следует Ответ MentalActionsImage и обычный Эффект.

После срабатывания Информационной функции информация добавляется к текущему информационному окружению в виде общей глобальной структуры mentalInfoStruct.

**5 стадия**

Набор правил реагирования позволяет не только прогнозировать развитие ситуации, но и провоцировать ее развитие в нужном Beast направлении. Например, подтолкнуть оператора совершить действия, решающие какие то проблемы Beast, например, улучшить его базовое состояние.