Характер в полкилобайта: куда встраивать веса в производство

Продолжение цикла, начатого «От жёсткой логики — к живой вероятности»: там — почему правила умирают на первом камне, в «Весах, которые пишут как код» — что веса стали текстом, в «Мы растим граф» — что граф строят из онтологии. Здесь спускаемся туда, где всё это будет стоять годами и молча: в щит, в станок, в контроллер. Вопрос простой: а есть ли весам место в настоящем производстве — и если есть, то какое именно, без дыма.

Пролог. Болото, о котором все знают

Откройте прошивку любого некрасивого, настоящего устройства — вентиляционной установки, насосной, подающего стола. Не учебную, а ту, что переписывали четыре человека за шесть лет.

Вы увидите болото. Пять сторожей тянут один исполнитель: перегрев говорит «стоп», просадка говорит «стоп», вибрация говорит «тише», кнопка говорит «ход», а протокол связи вообще молчит в своей таблице. Приоритеты зашиты в **порядок веток**: та, что выше в коде, побеждает. И это тот же палимпсест, что мы встретили в мире весов, — только здесь стёртые слои правят бал. Ветка, добавленная в 2021-м «быстро поправить цех», стоит выше ветки 2019-го и молча её переопределяет; через два года никто не помнит ни порядка, ни почему он именно такой.

Конечный автомат, которым пытаются вылечить болото, лечит одну болезнь и наживает другую: состояния — это декартово произведение всего со всем. Четыре фазы процесса, пять видов отказа, три режима — шестьдесят переходов, из них половина никогда не исполнялась и не проверялась. А матрица весов, которую все зовут «нейросетью и пугаются», приходит ровно в ту щель между болотом и автоматом, где её и нужно ставить. Разберём щель по линейке — и так же честно измерим, где ей не место.

Раздел 1. Щель между if и автоматом

Посмотрите, что общего у десятков реальных задач управления: **много слабых сигналов, один исполнитель, приоритеты конфликтуют, случайность отказа дорога**. Вибрация, ток и температура мотора — кто из них сегодня главный: стоп, предупреждение или просто тише? Пламя не пойман, концевик не дожат, давление гуляет — горелке что делать на этом такте? Линейка инфракрасных глаз у тележки — куда рулить, если «прямо» тоже вариант, но чуть хуже?

Это не вычисление. Это **арбитраж**. На каждом такте нужно выбрать одно из небольшого числа действий так, чтобы грубое перекрыло тонкое, а между собой тонкое честилось по мелочам. Ровно та структура, которую мы в прошлый раз записали лексикографикой: класс «стоп по неминуемой поломке» весит как 8192, «предупреждение» — 4096, и **сумма всех мелочей не может переорать старший класс — это арифметика, а не порядок строк**. Приоритет перестаёт быть следствием истории правок и становится числом. Число видно глазами, его правят одним патчем, и над ним можно спорить: «почему у тебя вибрация 3072, а просадка 4096?» — вопрос, которого в мире if вообще не существует.

Так что первое, что даёт матрица в щите: **приоритет как данные, а не как порядок**. Болото сворачивается в табличку, где каждая тревога — симптом, каждое действие — клетка выхода, и весь конфликт виден на одном листе.

Раздел 2. Честная граница: матрица не помнит

Было бы нечестно довести до читателя и не сказать, где подход кончается. Кончается он быстро и громко: **матрица не помнит**. Она не конечный автомат и не протокол. Парсинг по UART, «сначала прогрей три минуты, потом крути», счётчики, дебаунс кнопки — всё это последовательности, а у матрицы памяти нет вовсе: её ответ зависит только от того, что ей показали на этом такте.

Поэтому в живом контроллере матрица стоит не вместо кода, а на одном, зато самом гнилом участке. Рабочая архитектура такова:

код-энкодеры → матрица → гистерезис.

Энкодеры — обычный императивный код, самый скучный и надёжный — превращают сырые сенсоры в **симптомы**: «превышено три цикла подряд», «фаза прогрева идёт», «вибрация растёт второй такт». Там же живут таймеры, счётчики, память, весь протокол. Матрица делает ровно одно — решает, **какой рефлекс сейчас главный**. Гистерезис на выходе не даёт исполнительному реле дёргаться между равными. Никто не спорит с тем, что протокол — не матрица. Спорят только за кашу приоритетов — и эту кашу у контроллера забирают.

Где щель настоящая: сенсоры шумят, входов много, исполнитель один, приоритеты правят часто, и пропущенный отказ дороже простоя. Где её нет: логика на два порога (пишите if, будет честнее), большие пространства входов (руки держат несколько килобайт весов — не больше, и это потолок метода), и всё, где «автомат» означает протокол, а не выбор из трёх тревог.

Раздел 3. Две суперсилы, которых нет у болота

У прошивки на матрице есть два свойства, за которые её терпят даже циничные embedded-инженеры.

Первая — детерминизм по построению. Прогон графа из трёхсот с лишним умножений не содержит ни одной ветки. Ни одного «обычно миллисекунда, но вот если датчик чихнёт — зависит от мира». Худшее время хода известно **на этапе проекта**: на шестнадцатимегагерцовом чипе за доллар — полмиллисекунды на весь выбор, на сто тридцати трёх — в двадцать раз меньше, а между тактами контроллера — сотни тысяч тактов запаса, семь порядков. Для реального времени это не производительность. Это предсказуемость: худший случай ровно такой, каким его посчитали, — никогда не лучше и никогда хуже.

Вторая, и главная — характер становится данными. Вся воля машины — полкилобайта целых чисел. Правка «вибрация должна останавливать раньше» — это **одно число**, а не раскопка болота с проверкой, не сдвинулся ли чужой приоритет вместе с веткой. Тот же бинарник прошивки с другим файлом весов крутит другой характер: версии поведения живут в системе версий, откат поведения — откат файла, обновление по воздуху приносит **поведение, не трогая механику**. И прошивка при включении сама себя проверяет по эталонному набору позиций: мигнула «восемьдесят из восьмидесяти» — характер цел, та самая сеть, что собирали. Ветвистая логика так не умеет никогда: там доказательство корректности — покрытие условий, а условий экспонента.

Сложите две силы — и станет видно, для чего всё это тащат на железо не ради игры. Не «искусственный интеллект на камне», а **арбитраж с известным временем реакции и проверяемым характером**, который весит меньше, чем один ваш длинный if-elseif на двадцать условий.

Раздел 4. Горизонт без дыма — как всегда

Соблазн на этом месте — пообещать «нейросеть в каждый датчик». Не будет. Цикл всегда кончался честным разбором по этажам, разберём и этот.

Чего матрица в щите **не сделает**: не выучит новое поведение сама (обучения нет — правку приносит человек или модель-программист, и принимает её рефери); не заменит автомат процесса (памяти нет); не заслонит собой плохой датчик (энкодеры из мусора сделают мусорные симптомы аккуратно).

Что сделает — и это уже доказано на играх и матрицах знаний, а не обещано: малое пространство симптомов **перебирается целиком**. Триста восемьдесят комбинаций флагов — не экспонента, а таблица: прогоняем все, сверяем каждый выход с булевым двойником, и «0 ошибок» здесь значит буквально 0, на всех входах, навсегда, до следующей правки. Для устройства, которое должно проработать десять лет без присмотра, это другое качество надёжности: не «тестировали и вроде покрывает», а «перебрано и исчерпано».

А настоящая проверка метода впереди, и она не в тексте, а на станке: взять одну живую задачу с конечным набором симптомов — мотор, горелка, подающий стол, — пройти полный цикл от онто-каркаса («что существует: токи, фазы, тревоги; что из чего следует») через граф, компилятор и перебор — и посмотреть, выдержит ли методика контакт с шумом реальных сенсоров, с дребезгом, с «а давай ещё один датчик добавим». Методика говорит: добавление датчика = новый симптом = новые рёбра, а не новая архитектура. Посмотрим. Если выдержит — подход перестаёт быть нашей игрушкой и становится инструментом с инструкцией.

Эпилог. Ландшафт вместо палимпсеста

В начале цикла робот шёл по камням, и мы прятали каждый камень в очередную ветку. Потом веса стали текстом, текст — графом, а граф оказался тем самым ландшафтом, по которому поведение скатывается единственно возможным образом.

И вот производственная мораль, самая приземлённая из всех, какие мы тут выводили: прошивка болота — тоже палимпсест, просто его никто не счищал. Порядок веток — это отпечатки всех слоёв истории, застывшие в власть. Матрица не убирает историю — она делает её **прозрачной**: приоритет, который видно числом, можно обсуждать, править и откатывать, вместо того чтобы археологически раскапывать, почему тринадцатая строка перекрывает седьмую.

> *Болото — это не злые программисты. Болото — это приоритеты, спрятанные в порядке. Достаньте приоритеты в числа — и на месте болота окажется ландшафт.*

Продолжение и источник

• Начало цикла — «От жёсткой логики — к живой вероятности».

• Вторая и третья части — «От живой вероятности — к весам, которые пишут как код» и «Когда мы пишем веса — мы растим граф».

Идеи и эксперименты — автора (Telegram: @Tottop10, pro100.D); литературная и программная огранка — агент. CC BY 4.0.