

УДК 611.81 + 004.81 + 159.942

НИЗКОУРОВНЕВАЯ РЕАЛИЗАЦИЯ МОДЕЛИ НОРАДРЕНАЛИНОВОЙ ПОДСИСТЕМЫ НЕЙРОННОЙ ИМПУЛЬСНОЙ СЕТИ

В.В. Пищулин¹, М.О. Таланов²

Казанский (Приволжский) федеральный университет

¹dalv6666@gmail.com; ²max.talanov@gmail.com

Аннотация

Нор-адреналиновая подсистема играет важную роль в эмоциональной оценке и обратной связи так же, как и в принятии решений. Мы представляем программную систему для автоматической генерации кода на языке PyNEST, основываясь на высокоуровневом описании нейронных подсистем.

Ключевые слова: NEST, NeuCogAR, куб Лёвхейма, норадреналин

ВВЕДЕНИЕ

Исходя из эволюционной роли эмоций в Естественном Интеллекте, Искусственный интеллект (ИИ), не реализуя эмоциональные процессы, в действительности очень многое упускает. В ИИ невозможно воплотить многие феномены, которые связаны именно с эмоциональной картиной, эмоциональным состоянием человека [1].

Развитие когнитивных наук, в частности, когнитивной нейробиологии, позволяет двигаться дальше в вопросе реализации эмоциональных процессов в ИИ. Неврологи выявили, что мозг работает в эмоциональном цикле [2, 3], помимо того, они смогли локализовать часть мозга, отвечающую за сам цикл и его влияние на процессы принятия решений [4]. Актуальность темы подчеркивает американский учёный в области ИИ Марвин Мински, сооснователь лаборатории ИИ в Массачусетском технологическом институте, в его книге [5] частично раскрываются тема эмоциональных циклов и значимость эмоционального процесса для создания сильного ИИ с зачатками самосознания. Психоэмоциональные аспекты позволяют эмоциональным агентам регулировать вычислительную нагрузку и переключаться с задачи на задачу более осмысленно.

Благодаря результатам развития когнитивных наук, таких, как психология и когнитивная нейробиология, возможна новая степень детализации в углубленном понимании эмоций в целом. На основе теоретического открытия восьми основных эмоциональных элементов по Томкинсу [6], нейромодуляционной регуляции эмоционального цикла в участках головного мозга и развития математических моделей биологических нейронов имеется возможность проверить гипотезу, представленную эмоциональным кубом Лёвхейма [7]. Сбор информации о нейромодуляционных регуляциях эмоционального цикла в участках головного мозга даёт возможность установить пути нейромедиаторов в участках головного мозга, отображая их в виде схемы.

Целью работы было создание системы, позволяющей проецировать результат исследований нейромодуляционных регуляций эмоционального цикла в участках головного мозга, выраженный в схеме, максимально автоматизируя процесс трансформации на вычислительную архитектуру при проведении моделирования для проверки гипотез и анализа полученных результатов.

КОМПОНЕНТЫ СТРУКТУРЫ

В качестве графического представления информации использован ресурс draw.io, с помощью которого созданы графические представления отдельных частей головного мозга. Каждый элемент, представленный в графическом виде, имеет программную интерпретацию в модельной области. На рис. 1 изображен мезокортилимбический путь дофамина в головном мозге. Прямоугольники аметистового цвета являются участками головного мозга, в которых расположены рецепторы. Стрелками показана взаимосвязь между элементами.

На рис. 1 изображены участки головного мозга: **Cortex** – кора головного мозга, **VTA** – прилежащее ядро, **NAc** – вентральная область покрышки, **TPP** – педункулопонтинное тегментальное ядро и нейрорегуляторы, находящиеся в различных отделах мозга: **Glu** – глутаминовая кислота, **Ach** – ацетилхолин, **GABA** – γ-аминомасляная кислота, **DA** – дофамин.

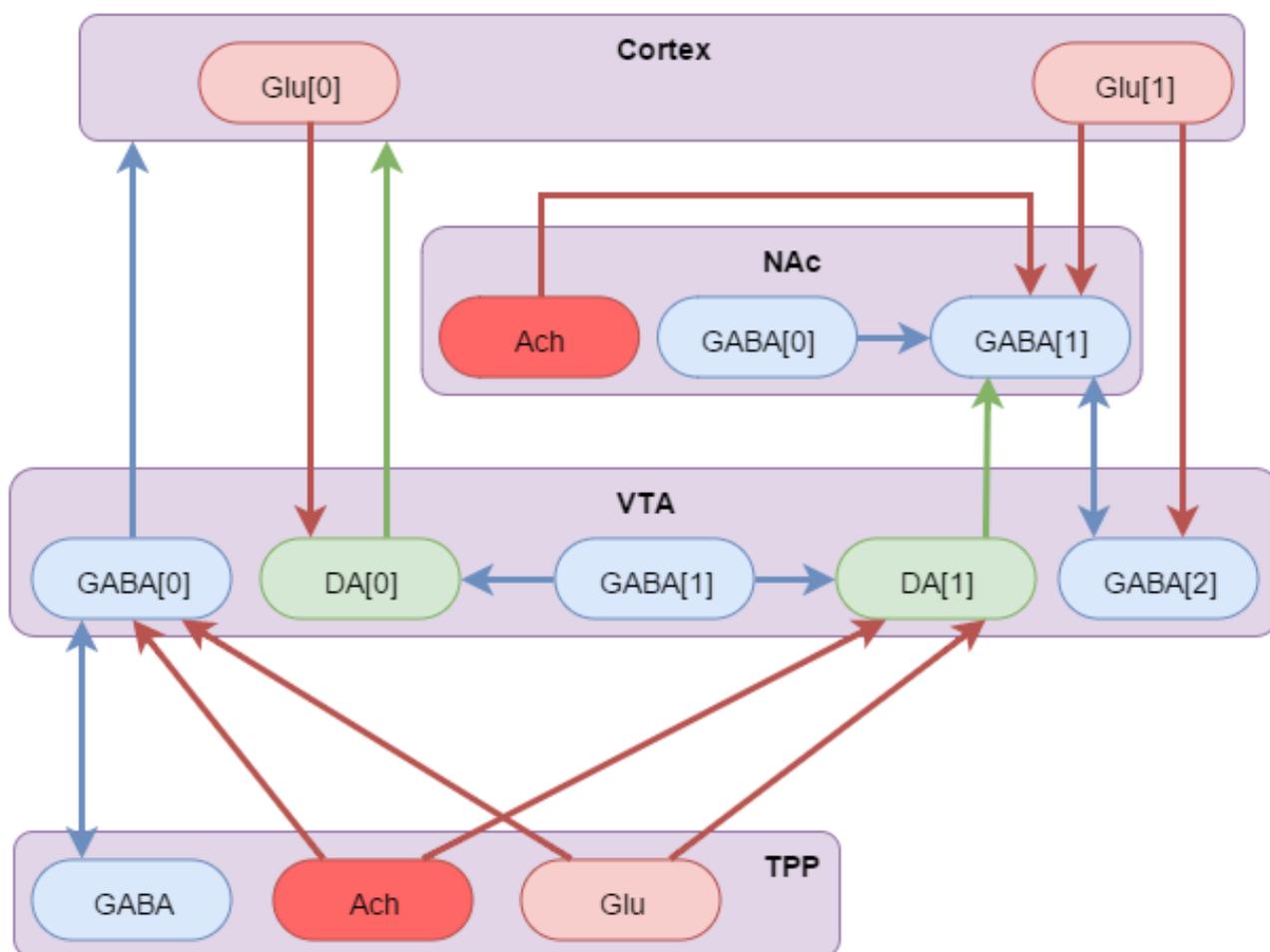


Рис. 1. Мезокортолимбический путь дофамина

Чтобы программный моделирующий комплекс смог распознать данное схематическое изображение участка головного мозга с последующим его моделированием, нужно представить данную схему в описанной структуре, а также составить легенду, позволяющую однозначно интерпретировать элементы схемы в модельную область. Ресурс draw.io имеет возможность экспортировать данную в схему в xml-файл.

Из схемы видно, что основными её компонентами являются участки головного мозга, рецепторы нейрорегуляторов и связи между этими рецепторами. Каждый элемент в XML-представлении имеет схожую нотацию, элемент, представляющий участок головного мозга VTA, изображен на рис. 2, рецептор DA[1] из VTA представлен на рис. 3, а связи между рецепторами – на рис. 4.

```
1 <mxCell id="58" value="VTA" style="..." parent="144"
2     vertex="1">
3     <mxGeometry x="11" y="4" width="229" height="206"
4         as="geometry"/>
5 </mxCell>
```

Рис. 2. XML-представление участка VTA

```
1 <mxCell id="62" value="DA[1]" style="plain-green"
2     parent="144" vertex="1">
3     <mxGeometry x="134.4545454545455"
4         y="147.409090909091"
5         width="94.09090909091"
6         height="38.181818181819"
7         as="geometry"/>
8 </mxCell>
```

Рис. 3. XML-представление рецептора DA [1]

```
1 <mxCell id="255"
2     style="edgeStyle=orthogonalEdgeStyle;rounded=0;"
3     parent="261"
4     source="203"
5     target="203"
6     edge="1">
7     <mxGeometry relative="1" as="geometry"/>
8 </mxCell>
```

Рис. 4. XML-представление связи между рецепторами

РАЗНОВИДНОСТИ СВЯЗЕЙ

На рис. 5 изображен интеграционный путь дофамина. Схема содержит большое количество взаимосвязей между элементами, выявим их.

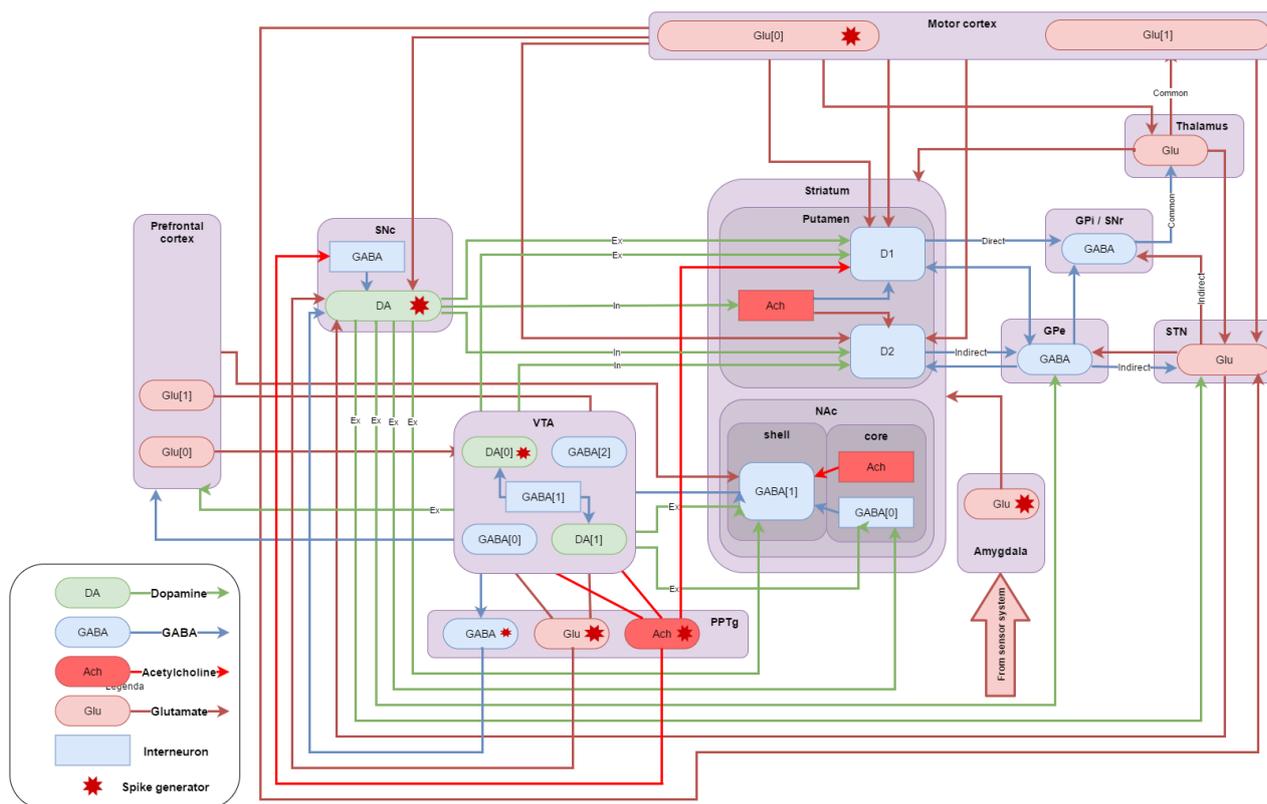


Рис. 5. Интеграционный путь дофамина

Связь рецептора и участка головного мозга

Рецептор может иметь связь с участком головного мозга. Это говорит о том, что данный рецептор имеет связь со всеми рецепторами из этого участка.

Связь участка головного мозга к рецептору

На рис. 7 выделена связь между Префронтальной корой (Prefrontal cortex) и γ -аминомасляной кислотой (GABA[1]) из прилежащего ядра. Это означает, что все рецепторы из Префронтальной коры имеют связь с рецептором γ -аминомасляной кислоты.

Связь участка головного мозга к головному мозгу

На рис. 5 такая связь отсутствует, но она может присутствовать на других схемах. Она будет означать, что все рецепторы из одного участка соединены со всеми рецепторами из другого.

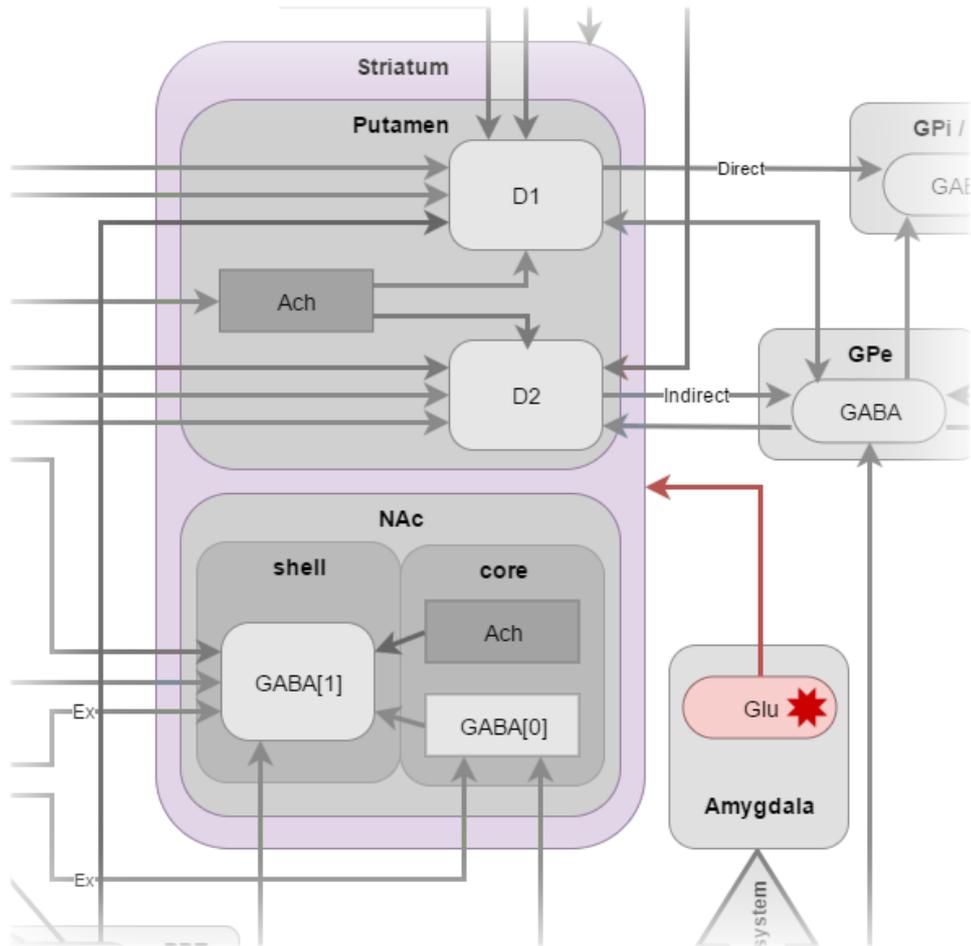


Рис. 6. Связь рецептора и участка головного мозга

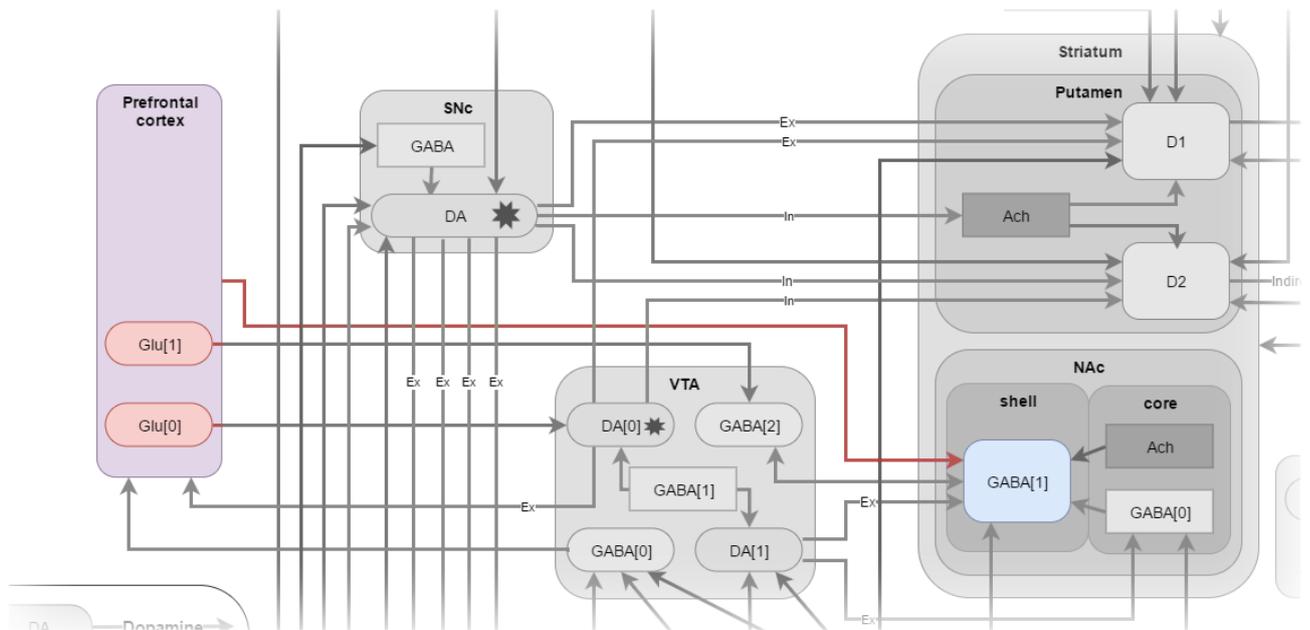


Рис. 7. Связь участка головного мозга и рецептора

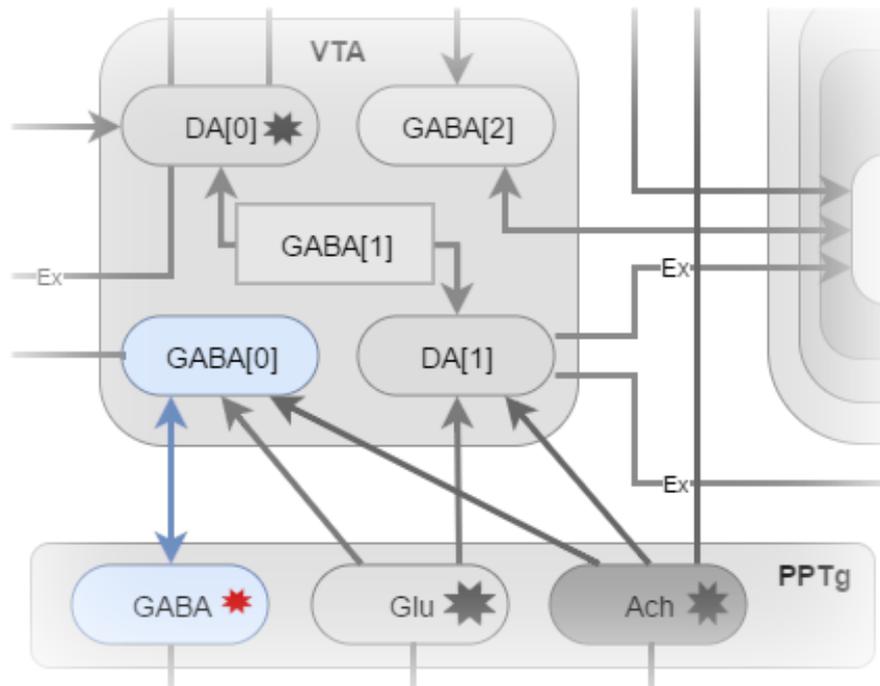


Рис. 8. Двухнаправленные связи

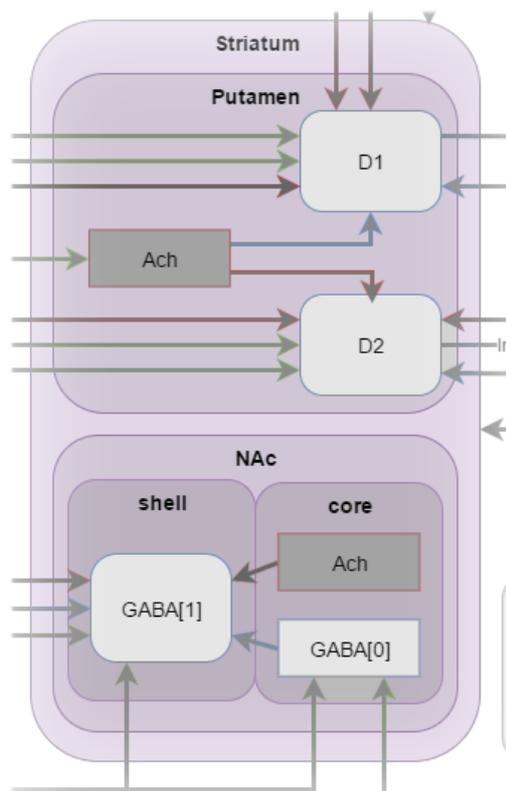


Рис. 9. Вложенные участки головного мозга

Двусторонние связи

На рис. 8 изображено двустороннее соединение рецепторов γ -аминомас-

ляных кислот (GABA[0] и GABA) из Вентральной области покрышки (VTA) и Педункулопонтинного тегментального ядра (PPTg). Это означает, что влияние рецепторов может быть двунаправленным, и это нужно учесть при создании моделей.

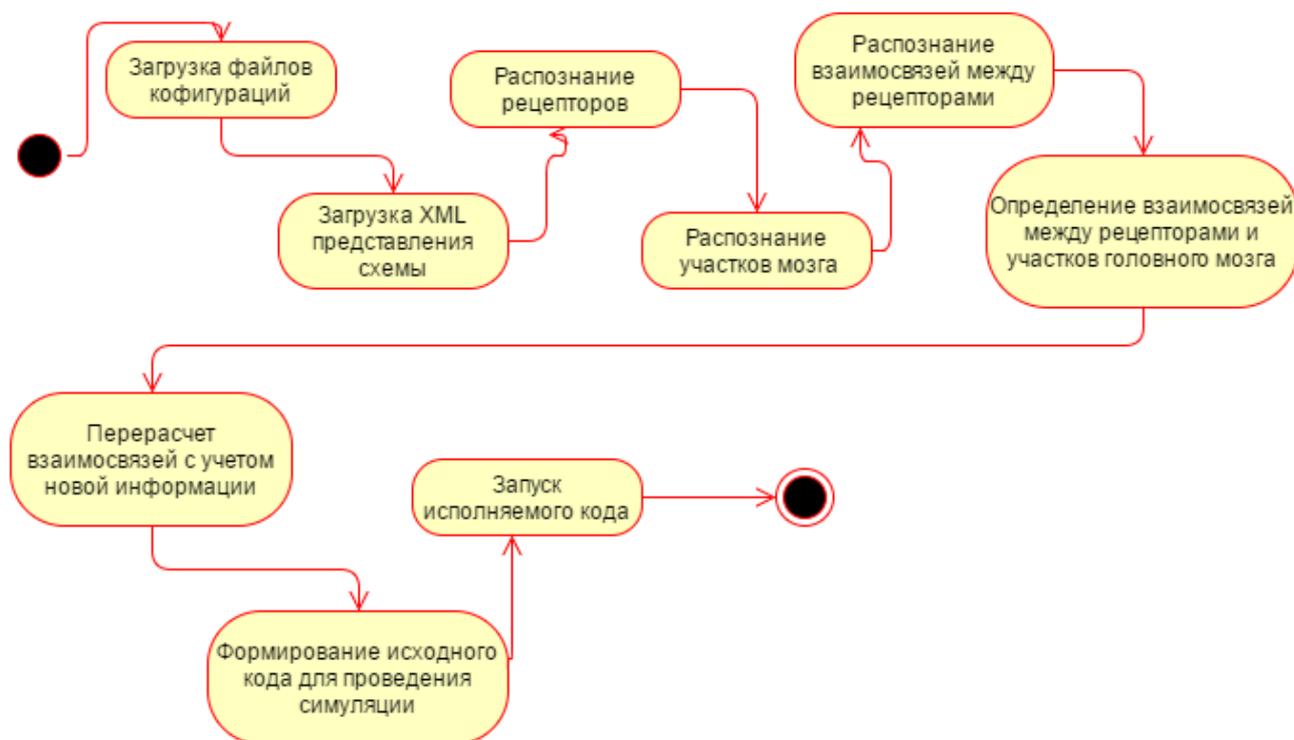


Рис. 10. Диаграмма деятельности

Вложенность участков мозга друг в друга

Участки головного мозга могут быть вложены друг в друга. Нам нужно обработать эту информацию, поскольку производится определение принадлежности рецептора конкретному участку мозга.

На рис. 9 изображено Полосатое тело (Striatum), которое содержит в себе Путамен (Putamen) и Префронтальную кору (NAc). Префронтальная кора состоит в свою очередь еще из несколько частей. Это означает, что при составлении модели данных нужно учитывать эту особенность вложений.

Основными связями в схемах могут выступать связи «рецептор–рецептор», «рецептор–участок мозга», «участок мозга–рецептор». Связи могут быть дву- или однонаправленными. Участки головного мозга обладают свойством вложенности.

ДИАГРАММА ДЕЯТЕЛЬНОСТИ

Определив основные компоненты описанной структуры и различные случаи их компоновки, составим диаграмму деятельности, которая наглядно представит алгоритм преобразования описанной структуры в исполняемый код для моделирования. Одним из первоначальных этапов является загрузка файлов конфигураций, основными из них являются `receptorsCount.properties`, `connectionWeight.properties`, `brainRegionsNames.list`, `receptorsNames.list` и `generatorConfig.json`.

```
1 ...
2 reticularformation_5HT=10
3 substantianigra_5HT=10
4 nucleusaccumbens_5HT=10
5 Thalamus_5HT=10
6 ...
```

Рис. 11. Структура файлов `properties`

Файлы `receptorsCount.properties` и `connectionWeight.properties` представляют собой обычные текстовые документы в формате «ключ–значение», где ключом является составное имя, состоящее из названий участка мозга и рецептора, соединенных с помощью символа подчеркивания. Значением является число. В файле `receptorsCount.properties` оно означает количество нейронов в данном рецепторе, а в файле `connectionWeight.properties` – все связи этого рецептора.

Файлы `brainRegionsNames.list`, `receptorsNames.list` используются для обеспечения гибкости системы. В файле `brainRegionsNames.list` хранятся все названия участков головного мозга, которые присутствуют на схеме, а в файле `receptorsNames.list` – все названия рецепторов.

```
1 ...
2 reticularformation
3 substantianigra
4 nucleusaccumbens
5 Thalamus
6 ...
```

Рис. 12. Структура файла `receptorsNames.list`

Файл `generatorConfig.json` – конфигурационный файл для настройки начального моделирования участка головного мозга.

```
1 {
2   "Thalamus_5HT": {
3     "name": "Thalamus_HT5",
4     "startTime": 400,
5     "stopTime": 600,
6     "rate": 250,
7     "coef_part": 1
8   }
9 }
```

Рис. 13. Структура файла generatorConfig.json

Загрузка XML-представления схемы

Для загрузки в XML используется технология JAXB (Java Architecture for XML Binding), которая позволяет ставить в соответствие Java-классы и XML файлы. Это дает возможность подгружать файл со схемой сразу в объекты классов программы и манипулировать далее с самими объектами.

Распознавание рецепторов и участков головного мозга

Оба эти этапа имеют общий принцип. XML-модель рецептора и участка головного мозга обладают полем "value", в котором хранится текстовое значение. Значение этого поля проверяется на наличие в словарях brainRegionsNames.list и receptorsNames.list. Если оно найдено в каком-то из словарей, то ему назначается модель в зависимости от словаря, в котором оно было найдено; в противном случае элемент игнорируется.

Распознавание взаимосвязей между рецепторами

На данном этапе определяются соединения между рецепторами. Отличительной чертой элементов взаимосвязей от элементов рецепторов или участков головного мозга является наличие у них двух дополнительных полей: "source" и "target". Это позволяет определять их однозначно. Стоит отметить, что эти поля могут ссылаться как на участок мозга, так и на отдельный рецептор, это вносит небольшую сложность, которую мы будем преодолевать чуть позже.

Определение взаимосвязей между рецепторами и участками головного мозга

На этапе распознавания взаимосвязей между рецепторами связи могут указывать на участок головного мозга; это значит, что связь указывает на все рецепторы в этом участке. Поэтому важно знать, какие рецепторы находятся внутри

участка. Для участков мозга без вложений это делается путем сравнения поля “parent” в элементах участка мозга и рецептора. Если значения у рецептора и участка мозга совпадают, это означает, что этот рецептор находится в данном участке.

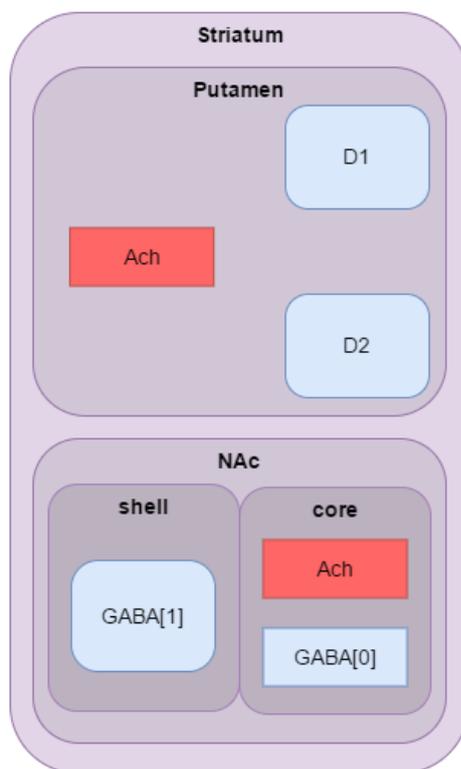


Рис. 14. Участок головного мозга Striatum

Иначе обстоит дело с многоуровневым вложением, например, с участком Striatum (рис. 14): все элементы имеют одинаковое значение поля “parent”, но Ach из участка core в NAc не принадлежит участку Putamen. Для решения этой проблемы используются поля элементов “x”, “y”, “height”, “wi “x”, “y” – это относительные координаты от верхнего левого угла главного родителя – Striatum. А “height”, “width” – длина и ширина элемента соответственно. Зная эти параметры, можно определить 4 крайние точки элемента. Исходя из них, определяем, лежит ли одна фигура в другой, и чем ближе расположены точки вложенной фигуры к фигуре, которая содержит вложение, тем более вероятно, что это фигура является родителем.

Формирование исходного кода для проведения симуляции



Рис. 15. Логотип фреймворка реалистичных сетей NEST

В качестве средства модуляции используется технология The Neural Simulation Tool (NEST) (рис. 15), которая базируется на следующей структуре (рис. 16).

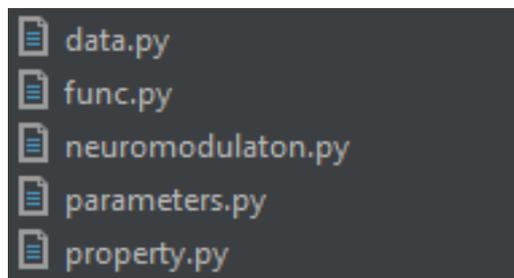


Рис. 16. Структура проекта NEUCOGAR

Основные файлы, которые подвержены изменению, – это data.py (файл, который хранит данные о каждом рецепторе: количестве, модели, типе) и neuromodulaton.py (файл, хранящий связи рецепторов между собой; он является исполняемым). Файл data.py (рис. 17) содержит кортежи с найденными участками мозга, а также ключи к этим кортежам, идентифицирующие рецепторы. А в кортежах хранятся параметры рецепторов, содержащихся в них: название, количество нейронов в этом рецепторе, модель нейрона, массив идентификаторов нейронов.

```
1 lateral_cortex = (  
2     {  
3         'Name': 'lateral_cortex[lateral_cortex_HT5]',  
4         'NN': 1000,  
5         'Model': 'iaf_psc_alpha',  
6         'IDs': nest.Create('iaf_psc_alpha', 1000)  
7     },  
8 )  
9 lateral_cortex_HT5 = 0
```

Рис. 17. Структура файла data.py

Файл neuromodulaton.py (рисунки 18–20) содержит связи между двумя рецепторами в следующем виде:

```
1 connect(  
2     lateral_cortex[lateral_cortex_HT5],  
3     DR[DR_HT5], syn_type=DĀ_ex,  
4     weight_coef=1.000000000  
5 )
```

Рис. 18. Структура файла neuromodulaton.py

Функция connect определена в файле func.py и позволяет соединять рецепторы между собой. Первый из параметров – сами рецепторы из файла определенных data.py, второй параметр (syn_type) – интерпретация связи между рецепторами, она указывается в схеме как цвет стрелки между рецепторами и определяется легендой самой схемы. Последний параметр (weight_coef) определяет силу связи между рецепторами и берётся из файла настройки connectionWeight.properties. Файл neuromodulaton.py содержит также настройки истоков генерации модуляции – генераторы в следующем виде.

Функция connect_generator (рис. 19) определена в файле func.py, она позволяет рецептору стать инициатором активности. Первым параметром указывается рецептор, определенный в файле data.py, вторым – время начала его активности, третьим – время окончания его активности, последними – коэффициенты для этого генератора.

```
1 connect_generator(  
2     Thalamus[Thalamus_HT5],  
3     startTime=400.000000000,  
4     stopTime=600.000000000,  
5     rate=250.000000000,  
6     coef_part=250.000000000  
7 )
```

Рис. 19. Структура файла neuromodulaton.py

Все настройки берутся из файла конфигурации generatorConfig.json. Последнее, что содержится в файле neuromodulaton.py, – это конфигурация измерительных моделей, которые позволяют получить результат проведенного моделирования. Функции connect_decetor и connect_multimeter определены в файле func.py,

они позволяют отслеживать состояние рецептора во время проведения моделирования.

```
1 connect_detector(lateral_cortex[lateral_cortex_HT5])  
2 connect_multimeter(lateral_cortex[lateral_cortex_HT5])
```

Рис. 20. Структура файла neuromodulaton.py

ИСПОЛЬЗОВАННЫЕ ТЕХНИЧЕСКИЕ СРЕДСТВА

В качестве языка программирования при реализации системы был выбран язык Java. Для моделирования во фреймворке NEST использован язык программирования Python. В качестве сред разработки для Java и Python выбраны IntelliJ Idea и PyCharm соответственно.

NEST является симулятором для нейронных сетевых моделей, который фокусируется на динамике, размере и структуре нейронных систем, а не на точной морфологии отдельных нейронов. Он идеально подходит для сетей нейронов любого размера: модель обработки информации, например, в визуальной или слуховой коре млекопитающих, модель динамики активности сети, например, ламинарные, корковые сети или сбалансированные случайные сети, модель обучения и пластичности. NEST представлен как открытый, расширяемый код, в него могут быть добавлены новые модели нейронов, синапсов и устройства [1]. NEST можно использовать и как интерпретатор для языка программирования Python (PyNEST), и как самостоятельное приложение [10].

PyNEST предоставляет собой набор команд для интерпретатора Python, который дает доступ к ядру моделирования. С помощью этих команд можно описать и запустить моделирование сети. Также имеется возможность дополнить PyNEST с помощью PyNN – симулятора независимого набора Python-команд для разработки и запуска нейронного моделирования. В то время, как описание моделирования происходит в среде Python, фактическое моделирование осуществляется в рамках оптимизированного ядра NEST, написанного на языке C++. Моделирование NEST пытается следовать логике электрофизиологического эксперимента, который происходит внутри компьютера, с той разницей, что исследуемая нейронная система должна быть определена экспериментатором [10].

JSON является открытым стандартом, который использует текст для передачи объектов данных, состоящих из пар «атрибут–значение». Это наиболее распространенный формат данных, который обладает широким спектром применения: используется для асинхронного взаимодействия браузера и сервера, а также в качестве конфигурационных файлов, в значительной степени заменяет XML, который используется в AJAX.

Extensible Markup Language (XML) является языком разметки, который определяет набор правил для кодирования документов в формате, одновременно удобном для восприятия человеком и машиной. Он определяется консорциумом W3C XML 1.0 и является свободным открытым стандартом.

Maven является инструментом для автоматизации сборки проектов, используется в основном для проектов Java. Он описывает структуру проекта и зависимости всего проекта. В отличие от предшествующих инструментов, таких, как Apache Ant, он использует соглашения для процедуры сборки. Файл XML описывает программное обеспечение сборки проекта, его зависимости от других внешних модулей и компонентов, порядок сборки, каталоги и необходимые плагины. Он поставляется с заранее определенными целями для выполнения таких задач, как компиляция кода и его упаковки. Maven динамически загружает Java-библиотеки и Maven-плагины из одного или нескольких хранилищ, таких, как Maven 2 Central Repository, и сохраняет их в локальном кэше [13].

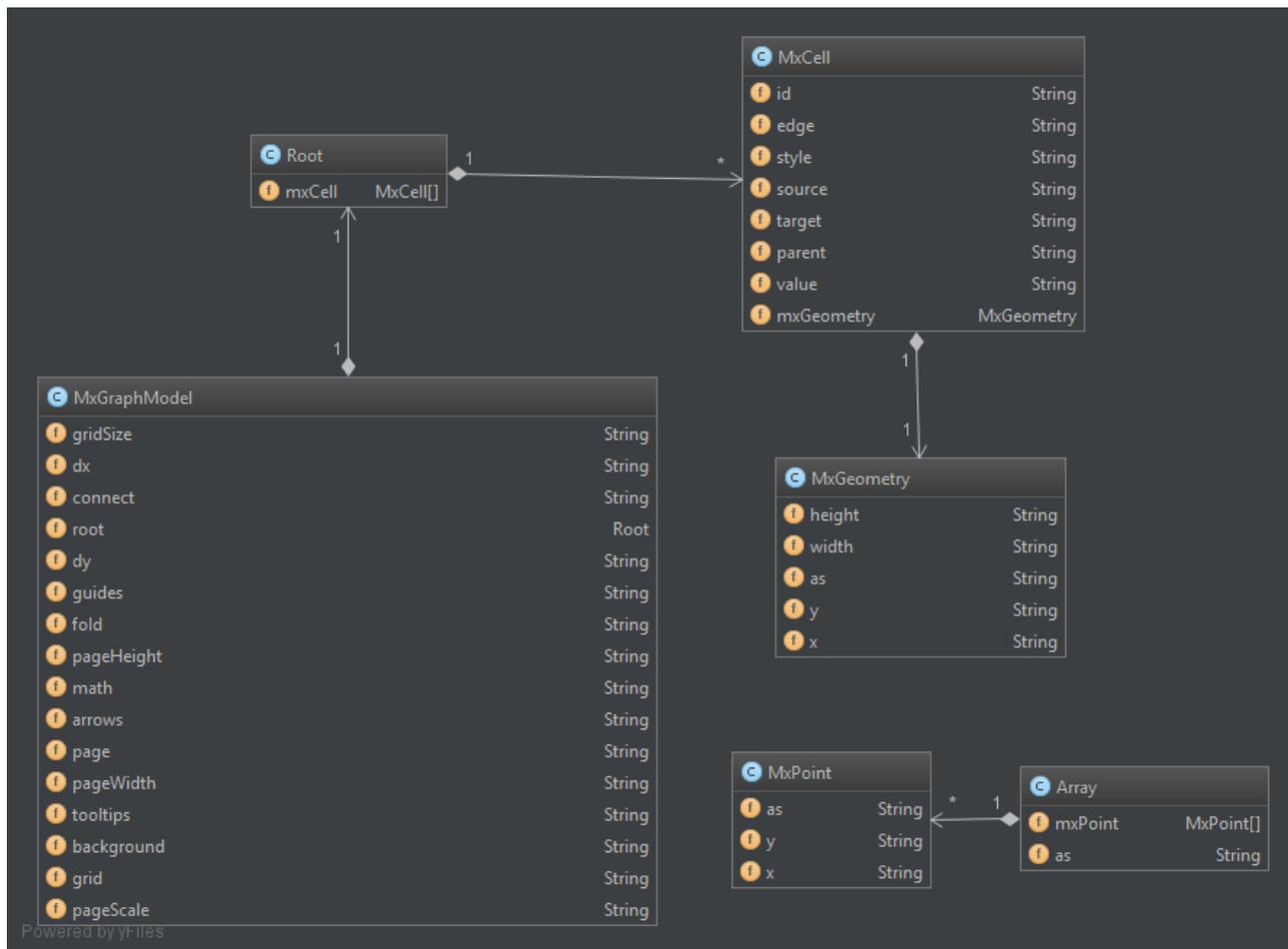


Рис. 21. Модели классов описанной структуры

РЕАЛИЗАЦИЯ

Первым этапом реализации является преобразования xml-представления схемы в модельную область java-программы. Такое преобразование можно выполнить с помощью технологии JAXB и воспользоваться онлайн сервисом. По входящей xml-модели были получены следующие структуры классов (рис. 21). Ключевую роль в этих моделях играет структура `MxCell`: именно в зависимости от значений полей этой структуры определяется тип элемента.

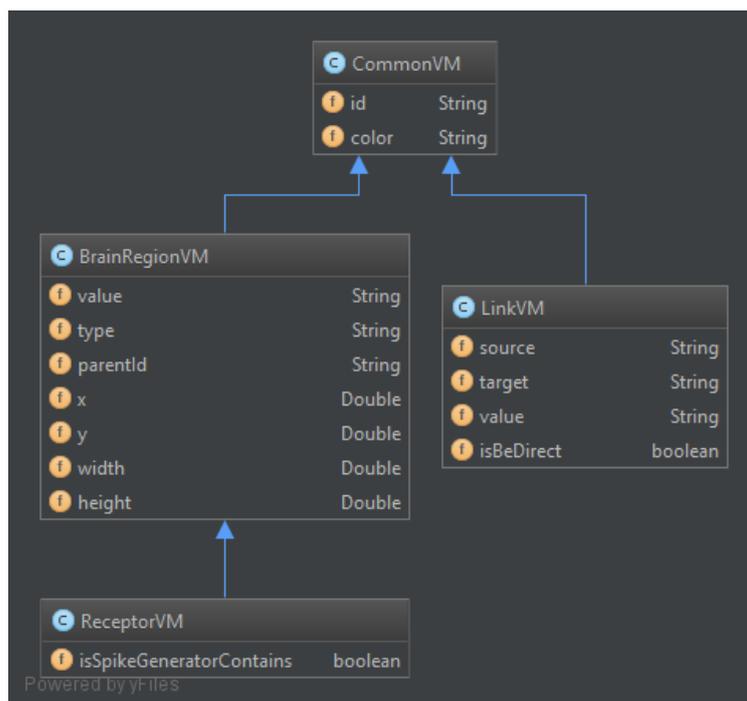


Рис. 22. Модели VM классов описанной структуры

Рассмотрим значение основных полей этой структуры: **id** – уникальный идентификатор, присваиваемый каждому элементу, **value** – текстовое значение элемента, **parent** – идентификатор родителя, **source** – идентификатор начала указателя, **target** – идентификатор конца указателя.

Если элемент содержит value, значит, это отдел головного мозга или группа рецепторов, и если у последних одинаково значение поля parent, то они находятся в одном отделе головного мозга. Элементы, содержащие поля source и target, являются указателями между элементами.

Для удобства работы и дальнейшей поддержки кода было принято решение преобразовывать модель MxCell в промежуточную структуру: **BrainRegionVM** – если это участок головного мозга, **LinkVM** – если это указатель между элементами, **ReceptorVM** – если это рецептор. Так как структуры имеют общие поля, они были вынесены в родительский класс CommonVM и унаследованы от него.

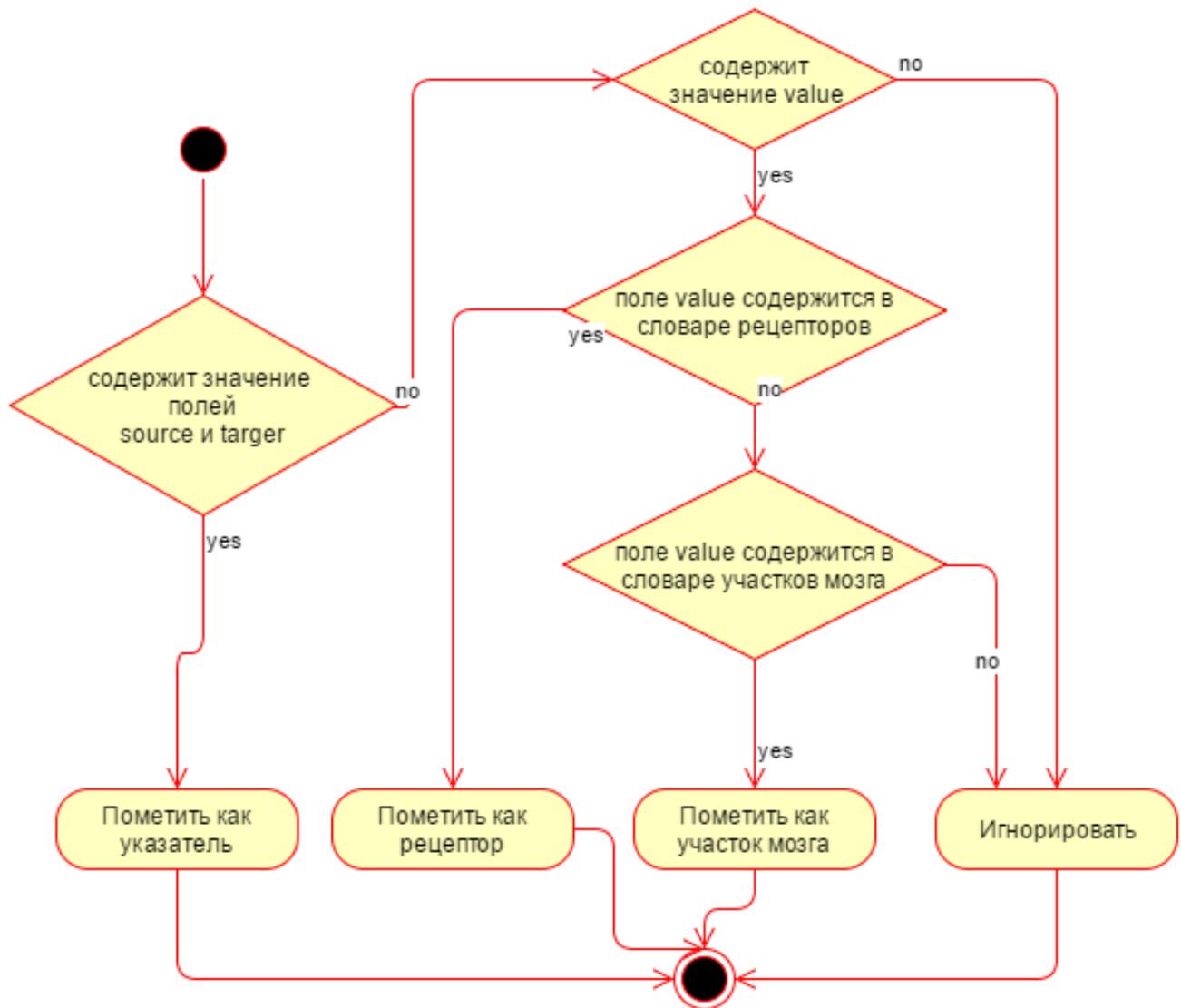


Рис. 23. Схема алгоритма определения элемента

Для однозначного преобразования структуры MxCell в вышеописанные модели был использован следующий алгоритм.

После того, как были определены основные элементы схемы, идентифицируются их взаимосвязи. Они проецируются в новую модель, которая в дальнейшем используется при моделировании.

Модель Receptor интерпретируется как рецептор со следующими полями: **id** – уникальный идентификатор, **BrainRegion** – ссылка на регион, в котором находится данный рецептор, **count** – количество нейронов в этом рецепторе, **type** – тип рецептора, **isSpikeGeneratorConnected** – флаг подключения генератора к этому рецептору, **connectedTo** – указатели на все рецепторы, с которым он соединен.

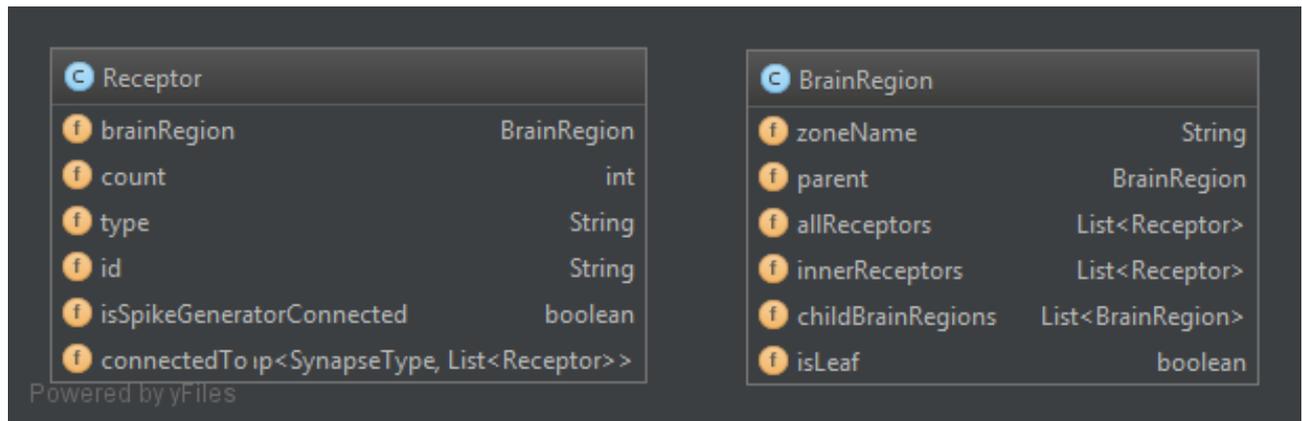


Рис. 24. Конечная модель данных

Модель BrainRegion интерпретируется как участок мозга со следующими полями: **zoneName** – имя участка, **parent** – ссылка на родителя, **allReceptors** – все рецепторы, расположенные внутри этого участка со вложениями другими участками, **innerReceptors** – рецепторы, расположенные только внутри этого участка, **childBrainRegions** – список указателей на вложенные участки, **isLeaf** – флаг окончания вложенности.

После создания списка всех описанных моделей формируем модели для проведения симуляции процесса внутри головного мозга с помощью технологии NEST. Чтобы сформировать эти данные, используем шаблоны (Рис. 25). Аналогичным образом заполняется шаблон файла neuromodulaton.py. Конечным результатом работы системы являются 5 выходных файлов, готовых для запуска на языке Python интерпретатором PyNest в фреймворке NEST.

```

1 import nest
2 import numpy as np
3
4 nest.ResetKernel()
5 nest.SetKernelStatus(
6     {'overwrite_files': True,
7      'local_num_threads': 4, 'resolution': 0.1
8     }
9 )
10
11 %1$2s

```

Рис. 25. Шаблон файла data.py

ЗАКЛЮЧЕНИЕ

Разработана система, позволяющая проецировать результат исследований нейромодуляционных регуляций эмоционального цикла в участках головного мозга, выраженный в схеме на вычислительную архитектуру для проверки гипотез и анализа полученных результатов в ходе проведения моделирования на фреймворке реалистичных сетей NEST.

СПИСОК ЛИТЕРАТУРЫ

1. *Максим Таланов*. Эмоциональные вычисления. URL: <http://postnauka.ru/video/45297>.
2. *Raine A., Yang Y.* The neuroanatomical bases of psychopathy. Handbook of Psychopathy. 2006. P. 278–295.
3. *Franks D.D.* The neuroscience of emotions. Handbook of the Sociology of Emotions. Springer, 2006. P. 38–62.
4. *Gottfried J.A., Haber S.N.* Neuroanatomy of reward: a view from the Ventral Striatum. 2011. 53 p.
5. *Minsky Marvin.* The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind. Simon and Schuster, 2007. P. 29.
6. *Kelly V.C.* A primer of affect psychology // The Art of Intimacy and the Hidden Challenge of Shame. 2009. P. 158–191.
7. *Lövheim H.* A new three-dimensional model for emotions and monoamine neurotransmitters // Medical Hypotheses. 2012. V. 78, No 2. P. 341–348.
8. Wikipedia Java. URL: <https://en.wikipedia.org/wiki/Java>.
9. Wikipedia Python. URL: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)).
10. *Gewaltig Marc-Oliver, Diesmann Markus.* NEST (NEural Simulation Tool) // Scholarpedia. 2007. V. 2, No 4. P. 1430. URL: [http://www.scholarpedia.org/article/NEST_\(NEural_Simulation_Tool\)](http://www.scholarpedia.org/article/NEST_(NEural_Simulation_Tool)).
11. Wikipedia JSON. URL: <https://en.wikipedia.org/wiki/JSON>.
12. Wikipedia XML. URL: <https://en.wikipedia.org/wiki/XML>.
13. Wikipedia Apache Maven. URL: https://en.wikipedia.org/wiki/Apache_Maven.

THE LOW LEVEL IMPLEMENTATION OF NORADRENALINE PATHWAYS VIA SPIKING NEURAL NETWORKS

V. Pishchulin¹, M. Talanov²

Kazan (Volga Region) Federal University

¹dalv6666@gmail.com; ²max.talanov@gmail.com

Abstract

The noradrenaline pathways plays important role in the emotional appraisal and feedback, as well as decision-making. We present the software system capable of automatic generation of PyNEST code based on high-level description of neuronal pathways.

Keywords: NEST, NeuCogAR, Lövheim's cube, noradrenaline

REFERENCES

1. Talanov M. Emotional computing. URL: <http://postnauka.ru/video/45297>.
2. Raine A., Yang Y. The neuroanatomical bases of psychopathy. Handbook of Psychopathy. 2006. P. 278–295.
3. Franks D.D. The neuroscience of emotions. Handbook of the Sociology of Emotions. Springer, 2006. P. 38–62.
4. Gottfried J.A., Haber S.N. Neuroanatomy of reward: a view from the Ventral Striatum. 2011. 53 p.
5. Minsky Marvin. The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind. Simon and Schuster, 2007. P. 29.
6. Kelly V.C. A primer of affect psychology // The Art of Intimacy and the Hidden Challenge of Shame. 2009. P. 158–191.
7. Lövheim H. A new three-dimensional model for emotions and monoamine neurotransmitters // Medical Hypotheses. 2012. V. 78, No 2. P. 341–348.
8. Wikipedia Java. URL: <https://en.wikipedia.org/wiki/Java>.
9. Wikipedia Python. URL: [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)).
10. Gewaltig Marc-Oliver, Diesmann Markus. NEST (NEural Simulation Tool) //

Scholarpedia. 2007. V. 2, No 4. P. 1430. URL: [http://www.scholarpedia.org/article/NEST_\(NEural_Simulation_Tool\)](http://www.scholarpedia.org/article/NEST_(NEural_Simulation_Tool)).

11. Wikipedia JSON. URL: <https://en.wikipedia.org/wiki/JSON>.
12. Wikipedia XML. URL: <https://en.wikipedia.org/wiki/XML>.
13. Wikipedia Apache Maven. URL: https://en.wikipedia.org/wiki/Apache_Maven.

СВЕДЕНИЯ ОБ АВТОРАХ



ПИЩУЛИН Владислав – выпускник 2016 года Высшей школы информационных технологий и информационных систем Казанского (Приволжского) федерального университета.

PISHCHULIN Vladislav, bachelor of Higher School Information technologies and systems of Kazan Federal University.

email: dalv6666@gmail.com



ТАЛАНОВ Максим Олегович – кандидат технических наук, руководитель Лаборатории машинного понимания Высшей школы информационных технологий и информационных систем Казанского (Приволжского) федерального университета (КФУ).

Maxim Olegovich TALANOV, PhD, head of Machine Cognition Lab in Higher Institute of Information Technology in Kazan (Volga region) Federal University.

email: max.talanov@gmail.com

Материал поступил в редакцию 15 мая 2016 года