

WEB-ПРИЛОЖЕНИЕ ДЛЯ ТЕСТИРОВАНИЙ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ ТЕОРИИ МУЛЬТИОПЕРАЦИЙ

Тодиков С. И.¹, ассистент, ✉ sergeytodikov@gmail.com, orcid.org/0000-0002-1249-6838
Показацкая А. В.¹, студент, avpokazatskaya@gmail.com

¹Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина), ул. Профессора Попова, 5, корп. 3, 197022, Санкт-Петербург, Россия

Аннотация

В работе представлено описание реализации двух из трех модулей интеллектуальной системы на основе теории мультиопераций. Для хранения данных используется реляционная модель. Для решения проблемы неопределенности применяются системы включений мультиопераций, элементы которых сохраняются в таблицах базы данных. Решение системы включений осуществляется путем подстановки вместо симптомов конкретных значений, которые определяются с помощью физиологических параметров пациента и модели симптомов. Такой подход позволяет получить объяснение результата на стороне клиента, не нагружая сервер приложения. Взаимодействие пользователя с системой происходит посредством человеко-машинного интерфейса. В статье также описан дополнительный функционал системы: дополнение или изменение системы уравнений, учет данных пользователя и другое. Разработанное web-приложение позволяет протестировать подход к построению систем принятия объяснимых решений на основе теории мультиопераций.

Ключевые слова: интеллектуальная система, свойство объяснимости, теория мультиопераций, система включений, web-приложение.

Цитирование: Тодиков С. И., Показацкая А. В. Web-приложение для тестирования интеллектуальной системы на основе теории мультиопераций // Компьютерные инструменты в образовании. 2024. № 1. С. 58–70. doi:10.32603/2071-2340-2024-1-58-70

1. ВВЕДЕНИЕ

На сегодняшний день разработка интеллектуальных систем имеет актуальный характер, так как во многих практических областях необходимы принятия решений, которые могут предоставить близкий к точному результат по определенным вопросам с объяснением того, как именно система пришла к окончательному результату. При создании таких систем особое внимание уделяется разработке модели представления знаний, которая способна более точно описывать состояния или события, включая рассмотрение неопределенных состояний. В настоящее время в большинстве случаев используются различные многозначные классические логики или нечеткие системы. Это в свою оче-

редь ведет к следующей проблеме: любое состояние неопределенности рассматривается между двумя логическими значениями — истина и ложь.

Другим важным аспектом при построении интеллектуальных систем является свойство объяснимости. В большинстве случаев интеллектуальные системы дают только конечный результат без всякого объяснения. Для конечных пользователей свойство объяснимости является существенным, так как при принятии ключевых решений им важно понимать, каким образом система пришла к окончательному результату.

В работе [1] для решения проблемы неопределенности предлагается использовать теорию мультиопераций и приводятся ее преимущества. Также в работе [1] приводится методика построения самообучающейся интеллектуальной системы со свойством объяснимости на основе теории мультиопераций.

В текущей работе описана реализация нескольких модулей интеллектуальной системы, на основе которых возможно провести первичное тестирование системы.

2. ЭЛЕМЕНТЫ ТЕОРИИ МУЛЬТИОПЕРАЦИЙ

Пусть 2^A — множество всех подмножеств A . Отображением из A^n в 2^A называется n -местной мультиоперацией на A [2]. Под рангом мультиоперации понимается мощность множества A . $k = |A|$.

Следуя [3], n -местную мультиоперацию f на множества $A = \{1, 2, 4\}$ можно представить как отображение:

$$f : \{1, 2, 4\}^n \rightarrow \{0, 1, 2, 3, 4, 5, 6, 7\}$$

используя следующую кодировку:

$$\{\emptyset\} \rightarrow 0; \{1\} \rightarrow 1; \{2\} \rightarrow 2; \{1, 2\} \rightarrow 3; \{4\} \rightarrow 4; \{1, 4\} \rightarrow 5; \{2, 4\} \rightarrow 6; \{1, 2, 4\} \rightarrow 7.$$

Также n -местную мультиоперацию f можно представить в виде вектора всех ее значений (a_1, \dots, a_{k^n}) , где $a_i \in \{0, 1, 2, 4\}$ и $a_i = f(2^{i_1}, \dots, 2^{i_n})$, а (i_1, \dots, i_n) есть представление $i - 1$ в системе счисления по основанию k n -разрядным числом.

Теория мультиопераций является подвидом многозначной логики. Только неопределенности, в отличие от классических многозначных логик или нечетких логик, интерпретируются иначе.

Например, в классической трехзначной логике есть три состояния, два из которых являются выделенными логическими значениями (истина и ложь) и третье, которое обозначается как неопределенность, — это есть состояние, которое может быть как истиной, так и ложью. Если увеличивать количество состояний, то будет увеличиваться только количество неопределенностей, но все эти неопределенности по различным правилам рассматриваются только между двумя логическими значениями — истина и ложь. И точно такая же картина в нечетких логиках, где состояния — это некие коэффициенты между 0 и 1.

В теории мультиопераций все элементы исходного множества являются логическими значениями. Если в качестве исходного множества рассматривать трехэлементное множество, то работа уже будет происходить с тремя логическими значениями, которые не зависят друг от друга. А под неопределенностями будут пониматься элементы, содержащиеся в образе более одного логического значения.

Следуя кодировке, которая представлена выше для мультиопераций ранга 3, на трехэлементном множестве получается 8 различных состояний. Если структурировать все

состояния, то получим 3 логических значения, 4 неопределенности и 1 бессмысленность (пустое множество):

- Логические элементы: $\{1\}, \{2\}, \{4\}$;
- Неопределенности: $\{1, 2\}, \{1, 4\}, \{2, 4\}, \{1, 2, 4\}$;
- Бессмысленность: $\{\emptyset\}$.

В неопределенностях рассматриваются все возможные варианты каждого логического элемента с другими логическими элементами.

Такой математический аппарат имеет следующие преимущества. Во-первых, информация, которая идет на вход аппарата и описывает некое событие или состояние объекта, может быть описана подробнее. Во-вторых, количество затрачиваемой памяти компьютера сокращается, так как для хранения 8 байтов информации в булевой логике необходимо три логических переменных, а в теории мультиопераций при ранге 3 достаточно одной логической переменной.

Определим терм с множеством неизвестных X и множеством констант K над множеством мультиопераций $F^n \subseteq M_A^k$ следующим образом:

- 1) если $y \in K \cup X$, то $t \equiv y$ — это терм и $U(t) = \{y\}$,
- 2) если $f \in F^n$ и t_1, \dots, t_n являются термами, то $t \equiv f(t_1, \dots, t_n)$ — это терм и $U(t) = U(t_1) \cup \dots \cup U(t_n)$,
- 3) если t_0 — терм, $U(t_0) = \{y_1, \dots, y_n\}$ и t_1, \dots, t_n являются термами, то $t \equiv t_0(t_1, \dots, t_n)$ — это терм и $U(t) = U(t_1) \cup \dots \cup U(t_n)$.

Общая формула системы включений с известными в теории мультиопераций имеет следующий вид [4]:

$$\begin{cases} t_1(\tilde{c}, \tilde{z}) \subseteq q_1(\tilde{c}, \tilde{z}), \\ \dots \\ t_m(\tilde{c}, \tilde{z}) \subseteq q_m(\tilde{c}, \tilde{z}), \end{cases}$$

где t_i, q_i термы с неизвестными \tilde{z} и константами \tilde{c} в $F^n \subseteq M_A^k$.

Решением системы включений является набор мультиопераций f_1, \dots, f_s таких, что все i выполняют включение при стандартном определении значения термов с помощью введенного оператора суперпозиции:

$$t_i(\tilde{c}, f_1(\tilde{c}), \dots, f_s(\tilde{c})) \subseteq q_i(\tilde{c}, f_1(\tilde{c}), \dots, f_s(\tilde{c})).$$

3. МЕТОДИКА ПОСТРОЕНИЯ САМООБУЧАЮЩЕЙСЯ ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ СО СВОЙСТВОМ ОБЪЯСНИМОСТИ

На основе полученных результатов в работах [4–6] в работе [1] приводится методика построения интеллектуальной системы медицинского характера со свойством объяснимости. В систему закладывается продукционная модель, которая основана на системах включений. Решение системы включений всегда можно проанализировать и дать объяснение полученного результата, что в свою очередь является свойством объяснимости.

Концепция системы включает три модуля.

Первый модуль — *клиент* — предназначен для взаимодействия оператора или эксперта с системой. Через него также можно получить объяснение решения, сформированного системой на основе текущей модели знаний.

Второй модуль — *сервер* — отвечает за работу с базой данных. Он обеспечивает функциональность, связанную с хранением и извлечением информации.

Третий модуль — *вычислительный* — занимается нахождением общего решения системы, а также анализирует полученное решение. Это позволяет определить, удовлетворяет ли новая заданная модель всем заранее заданным условиям, которые определены экспертами.

Заполнение базы знаний возможно не только с помощью оператора через клиента, но и с помощью ряда узконаправленных ИИ. На рисунке 1 представлена общая архитектура самообучающейся интеллектуальной системы со свойством объяснимости.

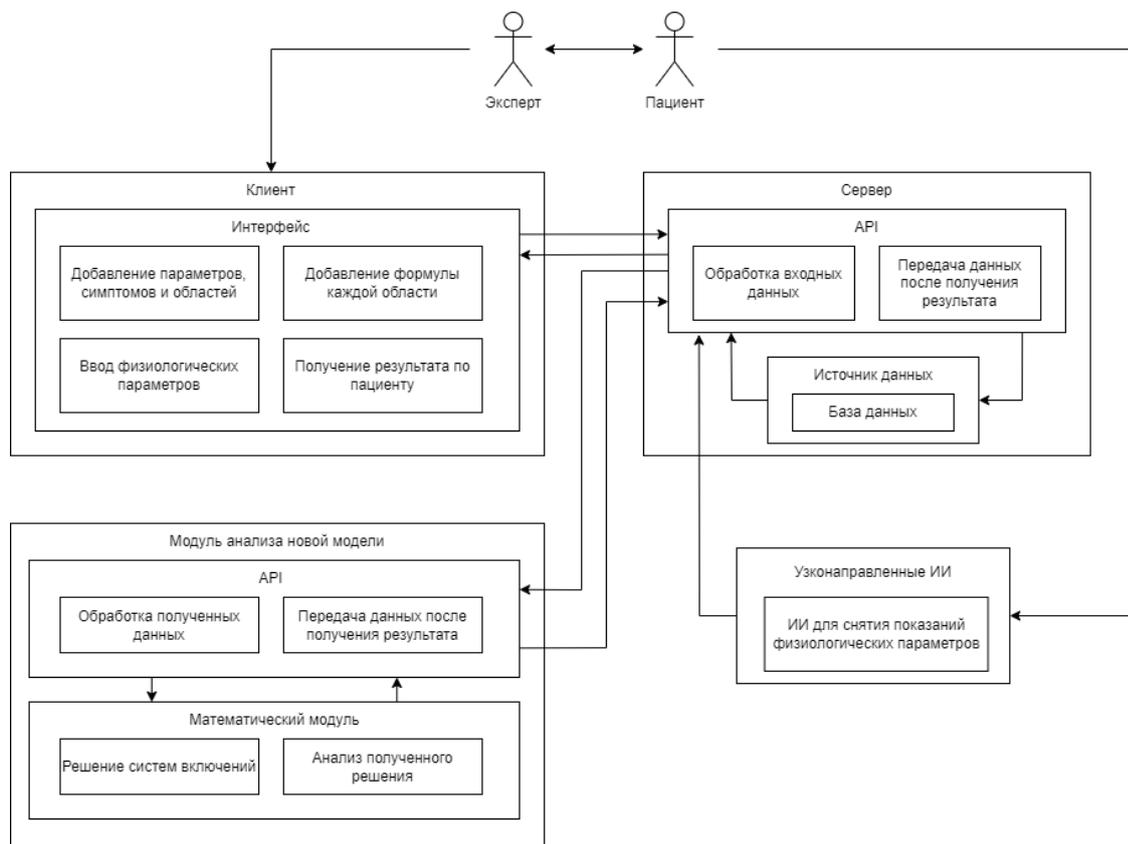


Рис. 1. Архитектура самообучающейся интеллектуальной системы со свойством объяснимости

Для того чтобы правильно интерпретировать входные и выходные данные, были разработаны две модели представления знаний на основе теории мультиопераций: модель симптомов и модель областей [1]:

Модель симптомов:

{∅} — эксперт не является квалифицированным для данного симптома;

{1} — симптома нет;

{2} — симптом есть;

{1, 2} — неопределенно (противоречие эксперта и аппарата);

{4} — аппарат дал невалидное значение;

{1, 4} — аппарат дал невалидное значение, эксперт на основе опыта делает предположение о том, что симптома нет;

{2, 4} — аппарат дал невалидное значение, эксперт на основе опыта делает предположение о том, что симптом есть;

{1, 2, 4} — нужны уточнения.

Под аппаратом понимается некое устройство, для снятия данных о физиологическом параметре, например, градусник, тонометр и т. д.

Модель областей:

$\{\emptyset\}$ — эксперт не является квалифицированным для d — не эксперт по текущей области;

$\{1\}$ — область не подходит;

$\{2\}$ — область подходит;

$\{1, 2\}$ — из-за противоречия эксперта и аппарата невозможно определить точное состояние области;

$\{4\}$ — невозможно определить состояние области, так как не о всех симптомах имеется информация;

$\{1, 4\}$ — на основе опыта эксперта делается вывод о том, что область не подходит;

$\{2, 4\}$ — на основе опыта эксперта делается вывод о том, что область подходит;

$\{1, 2, 4\}$ — для определения состояния области необходимы уточнения в симптомах.

Для того чтобы задавать системы включений, были определены следующие мультиоперации:

- Логическое И (\wedge) = (111 124 144);
- Логическое ИЛИ (\vee) = (124 222 424);
- Логическое отрицание ($-$) = (214).

Все вышеописанные мультиоперации образуют базис, на котором работает система включений.

В текущей работе приводится описание двух разработанных модулей: клиент и сервер.

4. ИСПОЛЬЗУЕМЫЕ ТЕХНОЛОГИИ

Взаимодействие клиента с программным интерфейсом приложения реализовано посредством архитектуры REST. Такой подход к построению архитектуры изолирует компоненты системы друг от друга, упрощает повторное использование уже существующих модулей и их тестирование, обеспечивает целостность данных за счёт объединения нескольких логических действий в одну транзакцию и повышает безопасность приложения, поскольку каждый компонент контролирует доступ к данным своего уровня.

4.1. База данных

В системе используется реляционная модель данных. Поэтому для хранения данных была выбрана реляционная база данных, которая реализована в СУБД Postgres. За счёт слаженной работы гибкой системы блокировок, планировщика задач и управления буферами и кешированием, а также невысоких требований к ресурсам система обеспечивает хорошую производительность и управляемость даже под сильной нагрузкой [7].

При проектировании базы данных были выделены четыре класса-сущности базы данных:

1. *Parameter* (параметр). Хранит название параметра, его максимальное и минимальное возможные значения, данные о вводе параметра, где 1 — ввод пользователем, 0 — ИИ.

2. *Symptom* (симптом). Хранит название симптома, максимальное и минимальное значение.
3. *Patient* (пациент). Хранит данные о пациенте (ФИО) и уникальный идентификатор.
4. *Area* (область). Хранит название области медицины и соответствующую формулу.

Пациенты имеют множество параметров, а параметры включают в себя различные симптомы.

Для каждой области свойственны наборы симптомов, связанных между собой с помощью логического оператора, по этой причине связывание таблиц с помощью связи «Многие ко многим» невозможно. Наиболее удачный вариант хранения логического выражения в базе данных — строка, в которой симптомы связаны между собой логическими операторами. Но при хранении строки возникает проблема сопоставления симптомов, поэтому запись логического выражения в таблицу имеет следующий вид: «s»+уникальный идентификатор симптома в базе данных. Для избежание дублирования формул в таблице областей, каждая формула перед записью в базу данных приводится к «стандартному виду». Под стандартным видом понимается приведенная к конъюнктивной нормальной форме формула и отсортированная по индексам симптомов в конъюнктуре. Так как при записи в базу данных есть ограничение на количество символов, а полученная формула может быть любой длины, приведенную к стандартному виду формулу преобразуем в битовую строку путем хеширования, уникальную для каждого набора символов, длины 64 символа.

На основе данной модели была построена диаграмма, в которой каждая сущность представляет таблицу базы данных. На рисунке 2 представлена сама диаграмма,

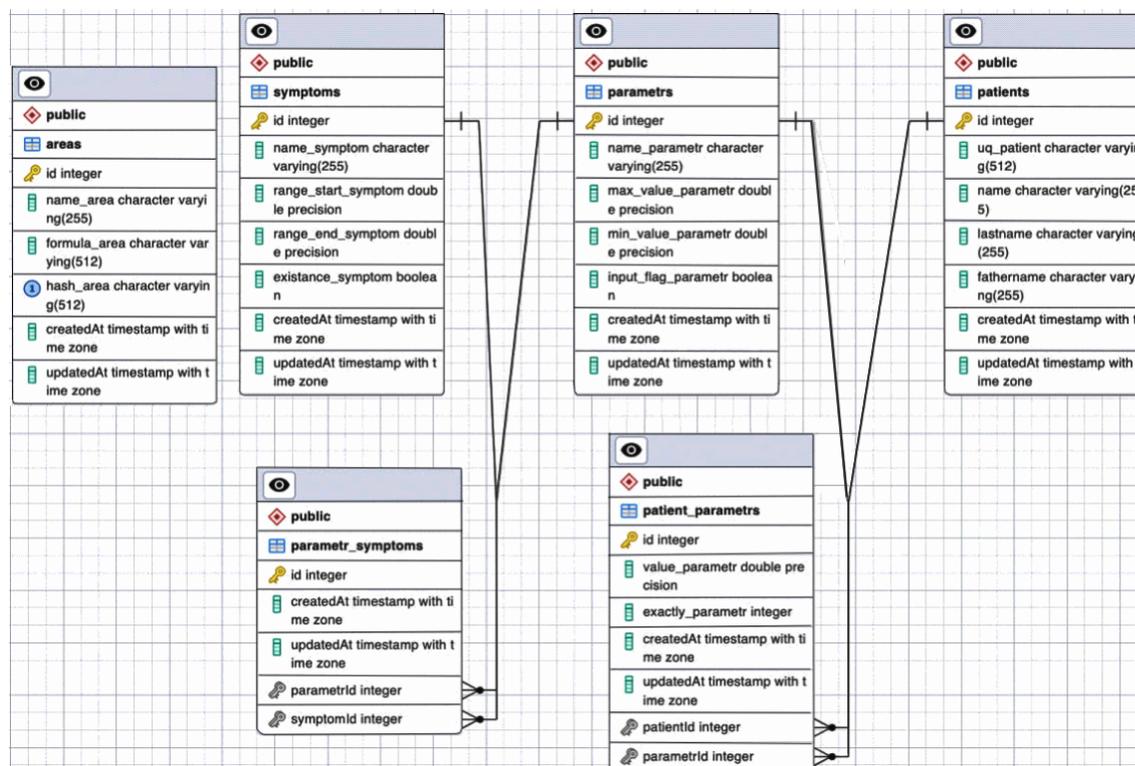


Рис. 2. Диаграмма базы данных

4.2. Сервер

Для разработки серверного слоя приложения был использован Node.js Express — это легкий и гибкий Node.js фреймворк, предоставляющий широкий набор функций для мобильных и веб-приложений [8].

Для организации взаимодействия между платформой Node.js и реляционными базами данными без использования специального языка запросов SQL используется Sequelize, предоставляющий работу с данными в терминах классов, а не таблиц, что позволяет существенно упростить реализацию приложения.

4.3. Клиент

Для реализации клиентской части веб-приложения выбран фреймворк Vue с использованием языка TypeScript.

Vue — это JavaScript-фреймворк для разработки пользовательских интерфейсов и одностраничных приложений. Он отличается высокой производительностью, простотой обучения и мощными функциями. Vue позволяет взаимодействовать с пользователями в интернете оптимальным и динамичным образом [9].

Взаимодействие пользователя с системой происходит посредством пользовательского интерфейса. Разработанный интерфейс предоставляет возможность заполнения таблиц базы данных сведениями о направлениях, физиологических параметрах, симптомах и пациентах.

5. ДЕМОНСТРАЦИЯ РАЗРАБОТАННЫХ МОДУЛЕЙ

Прежде чем начать работать с программой, требуется заполнить систему данными о симптомах и физиологических параметрах. После этого необходимо ввести для каждого направления формулу с характерными симптомами.

Добавление информации о новом направлении оформлено в виде калькулятора, где симптомы являются операндами, а логические функции операторами. Пользователю предоставляется выбор из уже введенных симптомов, если нужного симптома нет, пользователь имеет возможность добавления как нового физиологического параметра, так и нового симптома. На рисунке 3 представлен функционал для добавления нового физиологического параметра пациента.

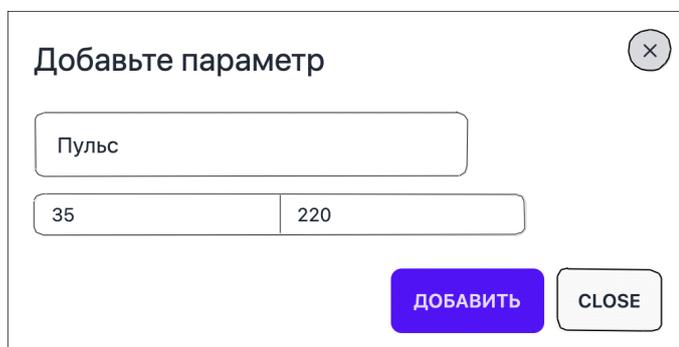


Рис. 3. Добавление нового физиологического параметра

На рисунке 4 представлен функционал для добавления нового симптома. На рисунке 5 представлено добавление информации о новом направлении.

The dialog box is titled "Добавьте новый СИМПТОМ" and contains the following elements:

- Header: "Добавьте новый СИМПТОМ" with a close button (X).
- Instruction: "Выберете параметр, к которому относится симптом".
- Radio button options:
 - Качество сна
 - Пульс
 - КГР
 - Мимика
 - Зрение
 - Температура
 - Давление
- Text input field: "Повышенный пульс".
- Range input field: "60" | "90".
- Buttons: "ДОБАВИТЬ" (blue) and "CLOSE" (grey).

Рис. 4. Добавление нового симптома

The dialog box is titled "Добавьте новое направление" and contains the following elements:

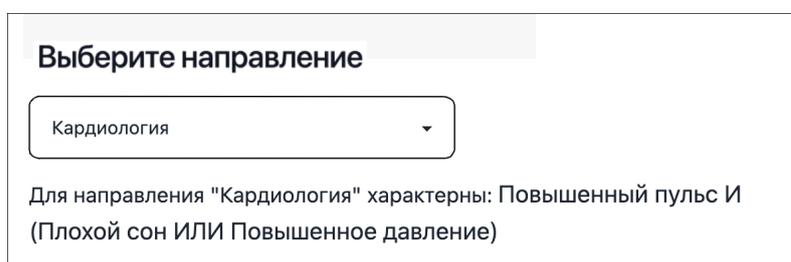
- Text input field: "Кардиология".
- Instruction: "Введите формулу".
- Text input field: "Повышенный пульс И (Плохой сон ИЛИ Повышенное давление)".
- Parameter selection area with buttons for:
 - Качество сна
 - Пульс
 - КГР
 - Мимика
 - Зрение
 - Температура
 - Давление
 - Повышенное давление
- Logic operators: "И", "ИЛИ", "НЕ".
- Grouping operators: "(", ")".
- Action buttons: "ОТМЕНИТЬ", "СТЕРЕТЬ", "ДОБАВИТЬ ПАРАМЕТР", "ДОБАВИТЬ СИМПТОМ".

Рис. 5. Добавление информации о новом направлении

Рассмотрим преобразование формулы в стандартный вид для отправления в базу данных на примере области «Кардиология». Перед заполнением данных о новой области необходимо добавить в систему физиологические параметры, характерные для области «Кардиология». Для данной области необходима информация о таких физиологических

параметрах, как «Температура», «Пульс» и «Качество сна». Характерные симптомы: «Повышенный пульс», «Плохой сон», «Повышенное давление».

Для области «Кардиология» из введенных симптомов можно составить формулу: «Повышенный пульс И (Плохой сон ИЛИ Повышенное давление)». При сохранении формула будет иметь вид: « $s1 \wedge (s7 \vee s2)$ », так как «Повышенный пульс» — симптом с индексом 1, «Плохой сон» — симптом с индексом 7, «Повышенное давление» — симптом с индексом 2, операторы: «И» — « \wedge », «ИЛИ» — « \vee ». После перевода в КНФ логическое выражение выглядит: « $((s2 \vee s7) \wedge s1)$ ». Следующим шагом убираем скобки, сортируем: « $s1 \wedge s2 \vee s7$ » и хешируем полученную строку: «\$2a10Ih4U0BX/lw3lg3TNLn5c4Ojj.dLYondWXUWkpwdjXEVBch5Mpx1M6». После перевода введенной формулы в хеш код, код сохраняется в базе, также в базе сохраняется запись « $s1 \wedge (s7 \vee s2)$ », чтобы просматривать записи о сохраненных направлениях. На рисунке 6 показано представление данных о направлении.



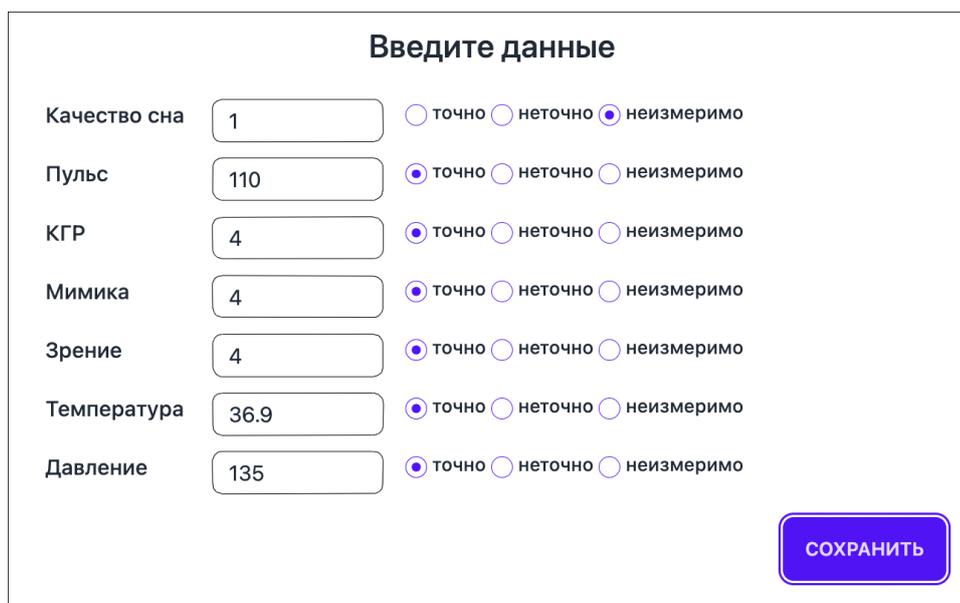
Выберите направление

Кардиология

Для направления "Кардиология" характерны: Повышенный пульс И (Плохой сон ИЛИ Повышенное давление)

Рис. 6. Представление данных о направлениях

Для того чтобы получить результат работы системы, необходимо заполнить форму с физиологическими параметрами и ввести необходимые значения. Если точно определить значение невозможно, необходимо выбрать соответствующее поле. На рисунке 7 представлена форма для заполнения физиологических параметров по пациенту.



Введите данные

Качество сна	<input type="text" value="1"/>	<input type="radio"/> точно <input type="radio"/> неточно <input checked="" type="radio"/> неизмеримо
Пульс	<input type="text" value="110"/>	<input checked="" type="radio"/> точно <input type="radio"/> неточно <input type="radio"/> неизмеримо
КГР	<input type="text" value="4"/>	<input checked="" type="radio"/> точно <input type="radio"/> неточно <input type="radio"/> неизмеримо
Мимика	<input type="text" value="4"/>	<input checked="" type="radio"/> точно <input type="radio"/> неточно <input type="radio"/> неизмеримо
Зрение	<input type="text" value="4"/>	<input checked="" type="radio"/> точно <input type="radio"/> неточно <input type="radio"/> неизмеримо
Температура	<input type="text" value="36.9"/>	<input checked="" type="radio"/> точно <input type="radio"/> неточно <input type="radio"/> неизмеримо
Давление	<input type="text" value="135"/>	<input checked="" type="radio"/> точно <input type="radio"/> неточно <input type="radio"/> неизмеримо

СОХРАНИТЬ

Рис. 7. Форма для заполнения физиологических параметров пациента

Для получения и объяснения результата на клиенте был разработан специальный модуль. Этот модуль был разработан специально на стороне клиента, так как объем вычислений в нем является небольшим, и современные устройства с легкостью смогут провести такие вычисления. Благодаря этому снимается лишняя нагрузка с сервера.

Суть этого модуля заключается в решении системы включений путем подстановки вместо симптомов конкретных значений, которые определяются с помощью физиологических параметров пациента и модели симптомов. Каждое включение в системе включений разбивается на обычное бинарное дерево, где нижний слой — это симптомы, а все верхние слои — это мультиоперации, которые включены в базис. Затем каждому симптому на основе физиологического параметра подставляется соответствующее значение из модели симптомов. Далее происходит решение бинарного дерева. После выполнения каждой мультиоперации и получения результата на определенном шагу модуль может сразу же проанализировать ответ и дать объяснение, почему был получен именно такой результат.

Для примера разберем область «Кардиология». Пусть у пациента имеется повышенный пульс, но на основе опыта эксперта, повышенное давление и хороший сон.

Возьмем включение для кардиологии: $s1 \wedge s7 \vee s2 \subseteq a_1$.

Заменим симптомы на конкретные значения: $6 \wedge 2 \vee 1 \subseteq a_1$,

Решим включение и получим следующий результат: $6 \subseteq a_1$, или $a_1 = 6$.

По модели областей делаем вывод о том, что кардиология подходит, если основываться на опыте эксперта. При этом на каждом шаге решения мы можем получить следующие объяснения:

$(2 \vee 1) = 2$, значит область подходит, так как повышенное давление имеется.

$6 \wedge 2 = 6$, значит область подходит, но уже на опыте эксперта, так как и повышенное давление и повышенный пульс имеется, но последний установлен на опыте эксперта.

На основе вышеописанного можно сделать вывод: область «Кардиология» может подходить, но только на основе опыта эксперта. На рисунке 8 продемонстрирован функционал для получения и объяснения результата для каждой рассматриваемой области.

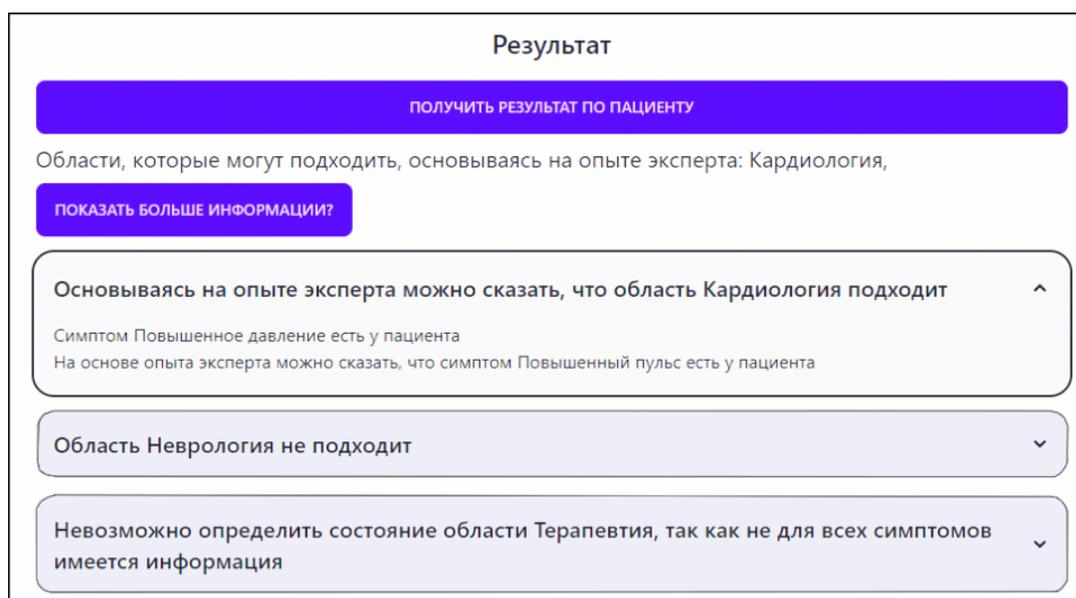


Рис. 8. Получение и объяснение результата

В этом случае необходима замена аппарата, который может измерить пульс. После того, как будут получены точные данные по пульсу, система сможет точно определить состояние области кардиология. Также это позволит узнать допустил ли эксперт ошибку или нет.

6. ЗАКЛЮЧЕНИЕ

Описанные в статье два из трех модулей для самообучающейся интеллектуальной системы подтверждают возможность и преимущества применения теории мультиопераций для систем принятия объяснимых решений. Проведенное тестирование приложения показало актуальность применения нового математического аппарата, основанного на теории мультиопераций. Представленные модули позволяют эксперту работать как с самой системой, так и с пациентами. В первом случае эксперт задает модели, которые основаны на системе включений. Во втором случае эксперт может передавать системе информацию о пациентах и получать развернутый ответ относительно каждой области, которая рассматривается в модели.

В дальнейшем планируется разработка третьего модуля — вычислительного. Данный модуль будет определять, подходит ли новая модель, которую задает эксперт, для системы или нет. Для этого планируется сначала получать общее решения для системы включений, а затем анализировать полученный результат по заранее заданным условиям.

Список литературы

1. Тодиков С. И. Возможность применения теории мультиопераций для построения самообучающихся систем искусственного интеллекта со свойством объяснимости // Материалы IV международной конференции по нейронным сетям и нейротехнологиям (NeuroNT'23). 2023. С. 105–108.
2. Перязев Н. А. Алгебры n -местных операций и мультиопераций // Материалы XV Международной конференции «Алгебра, теория чисел и дискретная геометрия: современные проблемы и приложения». Тула: ТПУ им Л.Н. Толстого, 2018. С. 113–116.
3. Перязев Н. А. Клоны, ко-клоны, гиперклоны и суперклоны // Учен. зап. Казан. гос. ун-та. Серия Физико-математические науки. 2009. Т. 151, кн. 2. С. 120–125.
4. Peryazev N. A. Systems of Inclusions with Unknowns in Multioperations // Bulletin of Irkutsk State University — Series Mathematics. 2022. Vol. 38. P. 112–123.
5. Перязев Н. А. Представление алгебр мультиопераций пространственными матрицами // Материалы XVIII Международной конференции «Алгебра, теория чисел и дискретная геометрия: современные проблемы и приложения». Тула: Изд-во ТПУ, 2020. С. 107–111.
6. Тодиков С. И. Автоматизация процесса решения систем включений с неизвестными в теории мультиопераций // Материалы 7-й международной конференции «Синтаксис и семантика логических систем». 2022. С. 53.
7. PostgreSQL — About. URL: <https://www.postgresql.org/about/> (дата обращения 08.02.2023).
8. Express.js. URL: <https://expressjs.com/> (дата обращения 08.02.2023).
9. Vue.js — Introduction. URL: <https://vuejs.org/guide/introduction.html> (дата обращения 08.02.2023).

Поступила в редакцию 11.12.2024, окончательный вариант — 08.02.2024.

Тодиков Сергей Игоревич, ассистент кафедры вычислительной техники СПбГЭТУ «ЛЭТИ»,
✉ sergeytodikov@gmail.com

Показацкая Арина Владимировна, студент кафедры вычислительной техники СПбГЭТУ
«ЛЭТИ», avpokazatskaya@gmail.com

Computer tools in education, 2024

№ 1: 58–70

<http://cte.eltech.ru>

[doi:10.32603/2071-2340-2024-1-58-70](https://doi.org/10.32603/2071-2340-2024-1-58-70)

Web Application for Testing Intelligent System Based on the Theory of Multioperations

Todikov S. I.¹, Assistant, ✉ sergeytodikov@gmail.com, orcid.org/0000-0002-1249-6838
Pokazatskaia A. V.¹, Student, avpokazatskaya@gmail.com

¹Saint Petersburg Electrotechnical University, 5, building 3, st. Professora Popova, 197022, Saint Petersburg, Russia

Abstract

The article describes the realization of two of three modules of an intelligent system which are based on the theory of multioperations. A relational model is used for keeping data. The multioperation inclusion system applies for solving the problem of uncertainty about which elements are stored in tables of the database. The solution of the system of inclusions is a partial solution where instead of symptoms specific values are substituted which are determined with the help of physical parameters of the patient and model of symptoms. This approach allows to get an explanation of the result on the client side without loading the server. Human-machine interface allows users to interact with the system. In the article the additional functionality of the system is also described: adding and changing the system of inclusions, recording user's data, etc. The developed web-application allows testing the approach to building an intelligent system with explanation property which is based on the theory of multioperations.

Keywords: *intelligent system, property of explanation, theory of multioperation, inclusion system, web-application.*

Citation: S. I. Todikov and A. V. Pokazatskaia, "Web Application for Testing Intelligent System Based on the Theory of Multioperations," *Computer tools in education*, no. 1, pp. 58–70, 2024 (in Russian); [doi:10.32603/2071-2340-2024-1-58-70](https://doi.org/10.32603/2071-2340-2024-1-58-70)

References

1. S. I. Todikov, "The Possibility of Applying the Theory of Multioperations to Build Self-learning Artificial Intelligence Systems with the Explainability Property," in *Proc. of 2023 IV International Conference on Neural Networks and Neurotechnologies (NeuroNT)*, pp. 56–59, 2023; [doi:10.1109/neuront58640.2023.10175856](https://doi.org/10.1109/neuront58640.2023.10175856)
2. N. A. Peryazev, "Algebras of n-ary Operations and Multioperations," in *Proc. of XV International Conference Algebra, Number Theory and Discrete Geometry: modern problems and applications, Tula State Pedagogical University, 28–31 May 2018, Tula, Russia*, pp. 113–116, 2018 (in Russian).
3. N. A. Peryazev, "Clones, Co-Clones, Hyperclones and Superclones," *Kazan. Gos. Univ. Uchen. Zap. Ser. Fiz.-Mat. Nauki*, vol. 151, no. 2, pp. 120–125, 2009 (in Russian).
4. N. A. Peryazev, "Systems of Inclusions with Unknowns in Multioperations," *Bulletin of Irkutsk State University - Series Mathematics*, vol. 38, pp. 112–123, 2022 (in Russian).
5. N. A. Peryazev, "Representation of algebras of multioperations by spatial matrices," in *Proc. of the XVIII Int. Conf. Algebra, number theory and, discrete geometry: modern problems, applications and problems of history, Tula State Pedagogical University, 23–26 Sep. 2020, Tula, Russia*, pp. 107–111, 2020 (in Russian).

6. S. I. Todikov, "Solving systems of inclusions with unknowns in multioperations," in *Proc. of the 7th Int. Conf. Syntax and semantics of logical systems, 1–5 Aug. 2022, Vladivostok, Russia*, p. 53, 2022 (in Russian).
7. The PostgreSQL Global Development Group, "PostgreSQL — About," in *www.postgresql.org*, 2024. [Online]. Available: <https://www.postgresql.org/about/>
8. StrongLoop et al., "Express 4.19.2 Fast, unopinionated, minimalist web framework for Node.js," in *expressjs.com*, 2024. [Online]. Available: <https://expressjs.com/>
9. E. You, "Vue.js — Introduction," in *vuejs.org*, 2024. [Online]. Available: <https://vuejs.org/guide/introduction.html>

Received 11-01-2024, the final version — 08-02-2024.

Sergey Todikov, Assistant, Department of Computer Science, St. Petersburg Electrotechnical University "LETI", ✉ sergeytodikov@gmail.com

Arina Pokazatskaia, Student, Department of Computer Science, St. Petersburg Electrotechnical University "LETI", avpokazatskaya@gmail.com