Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

|  |
| --- |
| Институт космических и информационных технологий |
| институт |
| Межинститутская базовая кафедра «Прикладная физика и космические технологии» |
| кафедра |

**ОТЧЕТ О ПРАКТИКЕ**

|  |
| --- |
|  |
| наименование дисциплины |
| Реализация продукционной экспертной системы в Android приложении |
| тема |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Преподаватель | |  |  |  |  | В.А. Углев |
|  | |  |  | подпись, дата |  | инициалы, фамилия |
| Студент | КИ20-01-3М | |  |  |  | Н.А. Болсуновский |
|  | номер группы, зачётной книжки | |  | подпись, дата |  | инициалы, фамилия |
|  |  | |  |  |  |  |

Красноярск 2022

**СОДЕРЖАНИЕ**

[Введение 2](#_Toc75279345)

[1 Общее описание…………………………………………………………………3](#_Toc75279346)

[2 Реализация системы……………….....................................................................5](#_Toc75279347)

[Заключение……………………………………………………………………….12](#_Toc75279352)

**ВВЕДЕНИЕ**

В ходе прохождения преддипломной практики необходимо было встроить возможности продукционной экспертной системы в Android приложение, разработанное в ходе прохождения первой практики.

**1 Общее описание**

Экспертными системами наиболее распространенного типа являются системы основанные на правилах. Правила, организованные в виде IF-THEN структур, называются продукционными правилам.

Продукционные правила, вместе с интерпретатором, который управляет их активизацией в зависимости от имеющихся фактов, составляют производственные модели представления и использования знаний в экспертных системах. Такие системы носят название продукционных.

В продукционных системах знания представлены в форме множественного правила, на основе которых формируются выводы, которые должны быть сделаны (или не сделаны) в разных ситуациях.

Выводы делаются на основе методов прямого или обратного логического вывода. Зависимости от метода логического вывода различают два вида продукционных систем: системы с прямым логическим выводом и системы с обратным логическим выводом.

Общая стратегия решения задач заключается в разбиении их на фрагменты, которые можно легче доказать. При этом, системы с прямым логическим выводом находятся под управлением фактов. Они начинают свою работу с известных начальных фактов и продолжают, используя правила для создания выводов или выполнения определенных действий.

Системы с обратной логическим выводом руководствуются гипотезами. Они начинают свою работу с гипотезы, или цели, которую пользователь пытается доказать и продолжают, отыскивая правила, которые позволят доказать правдивость гипотезы.

Широкое применение систем, основанных на продукционных правилах, обусловленное наличием в них следующих особенностей:

Модульная организация. Благодаря модульной организации упрощается представления знаний и расширение экспертной системы, наращивая ее возможности шаг за шагом.

Наличие средств объяснения. Продукционные экспертные системы с помощью правил позволяют легко создавать средства объяснения. Средство объяснения отслеживает последовательность активированных правил и, на этой основе, дает возможность восстановить ход рассуждений, которые привели к определенному выводу.

Наличие аналогии с познавательным процессом человека. Согласно результатам, полученным Ньюэллом и Саймоном, правила является естественным способом моделирования процесса решения задач человеком. Поэтому, в процессе выявления экспертных знаний, не возникают лишние сложности в объяснении экспертам структуры представления знаний, поскольку применяется простое их представление в виде правил IF-THEN.

**2 Реализация**

В качестве среды для разработки Android – приложения мной была выбрана Unity – межплатформенная среда разработки компьютерных игр, разработанная американской компанией Unity Technologies. Unity позволяет создавать приложения, работающие на более чем 25 различных платформах, включающих персональные компьютеры, игровые консоли, мобильные устройства, интернет-приложения и другие.

В данной практической работе мне необходимо интегрировать продукционную экспертную систему в приложение для поиска маршрута. Экспертная система по имеющимся данным о пассажире, должна порекомендовать следующую точку следования.

Для этого была произведем декомпозицию действий улетающего пассажира:

- первичный досмотр,

- регистрация пассажира,

- вторичный досмотри,

- посадка в самолет.

При выдаче рекомендации, система должна учитывать следующие факторы, касающиеся пассажира:

1. ФИО

2. Возраст

3. Вес

4. Номер телефона

5. Прописка

6. Наличие багажа

7. Вес багажа

8. Наличие доп. багажа (платный). Вес

9. Есть дети до 3х лет?

10. Тип места (эконом/бизнес)

11. Тип питания (рыба/курица/мясо/веган/кошерная)

12. Привит от корона-вируса?

13. Куда летит?

14. Время в полете

15. Есть инвалидность? Какая?

16. Есть почетные звания? Какие?

17. На каком языке говорит?

18. Везет домашнее животное? Какое?

19. Летал раньше?

20. Семейное положение

21. Багаж запакован?

22. Была судимость?

23. Есть неоплаченные налоги/штрафы/...?

24. Несет ответственность за кого-либо?

25. Находится в розыске какого-либо государства?

26. Прошел регистрацию?

27. Номер места на борту самолета

28. Номер багажной бирки

29. Предполетный досмотр пройден?

30. Номер гейта на посадку

Экспертная система представлена в виде xml – документа, имеющего следующую структура

# Модель FL в нотации xml

<FLMMODEL>

# Блок данных конфигурации

[-](file:///C:\Users\qwerty\Documents\PythonProjectsBook\flmdata.xml) <SETUP>

  <NAME> Название модели</NAME>

  <CAPTION>Описание модели</CAPTION>

  <COMMENTS>Комментарий</COMMENTS>

  <CREATEDATE>Дата создания</CREATEDATE>

  <MODDIFDATE>Дата последнего редактирования</MODDIFDATE>

  <USERNAME>Автор модели</USERNAME>

  <INPUTLEVELTERMSET>Количество терм-множеств на уровне входных данных</INPUTLEVELTERMSET>

  <PROCESSINGLEVELTERMSET> Количество терм-множеств на уровне обработки </PROCESSINGLEVELTERMSET>

  <OUTPUTLEVELTERMSET> Количество терм-множеств на уровне вывода </OUTPUTLEVELTERMSET>

  </SETUP>

# Блок данных о уровне ввода данных

[-](file:///C:\Users\qwerty\Documents\PythonProjectsBook\flmdata.xml) <INPUTLEVEL>

# Блок данных о первом терм множестве уровня ввода данных

[-](file:///C:\Users\qwerty\Documents\PythonProjectsBook\flmdata.xml) <SET\_1>

  <NAME>Название переменной первого терм – множества уровня ввода</NAME>

  <CAPTION> Название первого терм – множества </CAPTION>

  <TERMCOUNT>Количество термов в множестве</TERMCOUNT>

  <TERM1>Название первого терма</TERM1>

  <TERM2> Название второго терма </TERM2>

  <TERMN> Название N терма </TERMN>

</SET\_1>

  </INPUTLEVEL>

# Блок данных о уровнях обработки данных

[-](file:///C:\Users\qwerty\Documents\PythonProjectsBook\flmdata.xml) <PROCESSINGLEVEL>

# Блок данных о первом терм множестве уровня обработки

[-](file:///C:\Users\qwerty\Documents\PythonProjectsBook\flmdata.xml) <SET\_1>

  <NAME> Название переменной первого терм – множества уровня обработки </NAME>

  <CAPTION> Название первого терм – множества </CAPTION>

  <TERMCOUNT> Количество термов в множестве </TERMCOUNT>

  <TERM1> Название первого терма </TERM1>

  <TERM2> Название второго терма </TERM2>

  <TERMN> Название N терма </TERMN>

  </SET\_1>

  </PROCESSINGLEVEL>

# Блок данных о уровне вывода

[-](file:///C:\Users\qwerty\Documents\PythonProjectsBook\flmdata.xml) <OUTPUTLEVEL>

# Блок данных о первом терм множестве уровня вывода

[-](file:///C:\Users\qwerty\Documents\PythonProjectsBook\flmdata.xml) <SET\_1>

  <NAME> Название переменной первого терм – множества уровня вывода </NAME>

  <CAPTION> Название первого терм – множества </CAPTION>

  <TERMCOUNT> Количество термов в множестве </TERMCOUNT>

  <TERM1> Название первого терма </TERM1>

  <TERM2> Название второго терма </TERM2>

  <TERMN> Название N терма </TERMN>

  </SET\_1>

  </OUTPUTLEVEL>

# Блок данных о правилах

[-](file:///C:\Users\qwerty\Documents\PythonProjectsBook\flmdata.xml) <RULES>

  <RULES\_LEVELS\_COUNT>Правила между сколькими уровнями</RULES\_LEVELS\_COUNT>

  <RULES\_LEVELS1\_COUNT>Сколько подуровней правил на уровне 1 </RULES\_LEVELS1\_COUNT>

  <RULES\_LEVELS2\_COUNT> Сколько подуровней правил на уровне 2 </RULES\_LEVELS2\_COUNT>

# Блок данных о первом уровне правил

[-](file:///C:\Users\qwerty\Documents\PythonProjectsBook\flmdata.xml) <RULES\_LEVEL\_1> # Правила уровня 1

[-](file:///C:\Users\qwerty\Documents\PythonProjectsBook\flmdata.xml) <RULES\_LEVEL\_1\_1> # Правила подуровня\_1 уровня\_1

  <RULES\_COUNT>Сколько правил на подуровне 1\_1</RULES\_COUNT>

# Пример правила: условие1\_условие2\_условие3..решение\_коэффициент уверенности

  <RULE\_1>1\_1\_1..2\_1</RULE\_1>

  <RULE\_2>1\_1\_2..3\_0.9</RULE\_2>

  <RULE\_3>1\_1\_3..1\_0.9</RULE\_3>

  <RULE\_4>1\_2\_1..3\_0.8</RULE\_4>

  <RULE\_5>1\_2\_2..3\_0.8</RULE\_5>

  </RULES\_LEVEL\_1\_1>

  </RULES\_LEVEL\_1>

# Блок данных о втором уровне правил

[-](file:///C:\Users\qwerty\Documents\PythonProjectsBook\flmdata.xml) <RULES\_LEVEL\_2>

[-](file:///C:\Users\qwerty\Documents\PythonProjectsBook\flmdata.xml) <RULES\_LEVEL\_2\_1>

  <RULES\_COUNT> Сколько правил на подуровне 2\_1</RULES\_COUNT>

  <RULE\_1>1\_1..8\_1</RULE\_1>

  <RULE\_2>1\_2..5\_0.8</RULE\_2>

  <RULE\_3>1\_3..6\_0.9</RULE\_3>

  </RULES\_LEVEL\_2\_1>

  </RULES\_LEVEL\_2>

  </RULES>

# Блок данных о связях между уровнями

[-](file:///C:\Users\qwerty\Documents\PythonProjectsBook\flmdata.xml) <CONNECTIONS> # Тут задаются связи между уровнями

  <LEVELS\_COUNT>Сколько уровней связей</LEVELS\_COUNT>

# Блок данных о связях между первым и вторым уровнями

[-](file:///C:\Users\qwerty\Documents\PythonProjectsBook\flmdata.xml) <LEVEL\_1>

  <CONNECTIONS\_COUNT> Сколько связей на уровне 1 </CONNECTIONS\_COUNT>

  <TERM\_1\_1\_1>1\_1</TERM\_1\_1\_1>

  <TERM\_1\_1\_2>1\_1</TERM\_1\_1\_2>

# Пример связи: терм из уровня 1 (входной)\_1(1 терм-множество)\_3(3 терм) соединен с терм – множеством 1 и 2 уровня 2 (1 – есть связь/ 0 – связи нет )

  <TERM\_1\_1\_3>1\_1</TERM\_1\_1\_3>

  </LEVEL\_1>

# Блок данных о связях между вторым и третьим уровнями

[-](file:///C:\Users\qwerty\Documents\PythonProjectsBook\flmdata.xml) <LEVEL\_2>

  <CONNECTIONS\_COUNT> Сколько связей на уровне 2 </CONNECTIONS\_COUNT>

  <TERM\_2\_1\_1>1</TERM\_2\_1\_1>

  <TERM\_2\_1\_2>1</TERM\_2\_1\_2>

# Пример связи: терм из уровня 2 (обработки)\_1(1 терм-множество)\_3(3 терм) соединен с терм – множеством 1 уровня 3 (вывода )

  <TERM\_2\_1\_3>1</TERM\_2\_1\_3>

  </LEVEL\_2>

  </CONNECTIONS>

  </FLMMODEL>

Для того чтобы считать данные о экспертной системе, создадим скрипт ES, и добавим следующую функцию xmlParsing ()(рис. 1)

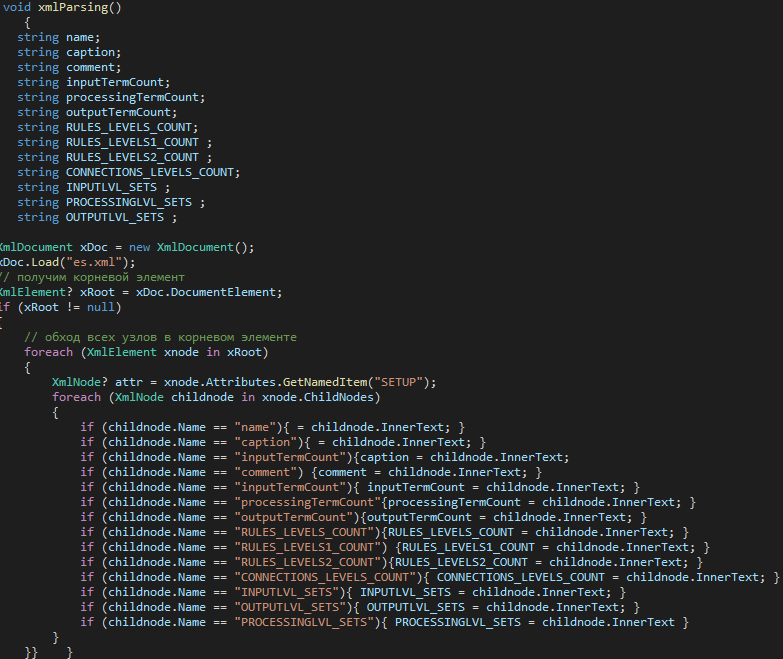


Рисунок 1 – Листинг функции ES.xmlParsing().

После того, как как данные о экспертной системе загружены, система должна подгрузить данные о пассажире из БД, что бы корректно дать рекомендацию, в случае с улетающим пассажиром, информация должна быть следующей:

- был ли пройден первичный досмотр,

- была ли пройдена регистрация пассажира,

- была ли пройдена регистрация багажа,

- был ли пройден вторичный досмотр,

- была ли пройдена посадка в самолет.

После определения ранее пройденных операций, физического состояния пассажира, экспертная система должна рекомендовать точку для дальнейшего следования, после чего, ее координаты должны быть переданы в систему построения маршрута

**Выводы**

В ходе прохождения преддипломной практики, мной были изучены правила построения продукционных экспертных систем.

Так же был внедрен модуль обработки экспертных систем в раннее разработанное приложение построителя маршрутов, в процессе разработки мной были освоены навыки:

- работы в среде разработки Unity,

- программирования на языке С#.