



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика, искусственный интеллект и системы управления

КАФЕДРА Системы обработки информации и управления

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

***Автоматизированное создание миварной
базы знаний на основе наборов формул***

Студент ИУ5-33М
(Группа)

(Подпись, дата)

(И.О.Фамилия)

Пасатюк А.Д.

Руководитель

(Подпись, дата)

Гапанюк Ю.Е.
(И.О.Фамилия)

2023 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой ИУ5
(Индекс)
_____ Терехов В.И.
(И.О.Фамилия)
« _____ » _____ 2023 г.

З А Д А Н И Е
на выполнение научно-исследовательской работы

по теме Автоматизированное создание миварной базы знаний на основе наборов формул

Студент группы ИУ5-33М

Пасатюк Александра Дмитриевна
(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)

Источник тематики (кафедра, предприятие, НИР) _____

График выполнения НИР: 25% к ____ нед., 50% к ____ нед., 75% к ____ нед., 100% к ____ нед.

Техническое задание Проанализировать структуру миварной базы знаний, определить формат и структуру входных данных для автоматизированной системы создания миварной базы знаний на основе наборов формул

Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на _____ листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

Дата выдачи задания « ____ » _____ 2023 г.

Руководитель НИР

(Подпись, дата)

Гапанюк Ю.Е.
(И.О.Фамилия)

Студент

(Подпись, дата)

Пасатюк А.Д.
(И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

Оглавление

Введение.....	4
Описание структуры базы знаний.....	5
Определение формата и структуры входных данных	10
Программные средства для обработки данных	13
Заключение	15
Список источников	16

Введение

Миварные технологии логического искусственного интеллекта (ЛИИ) [1] позволяют находить решение с линейной вычислительной сложностью для задач в форматах продукционных сетей «если – то» или описания бизнес-процессов в формате «вход; выход; действие». При этом создание самой базы знаний может занимать большое количество времени.

Чтобы ускорить процесс создания базы знаний, предлагается автоматизировать данный процесс. И тогда эксперту не придется тратить время на изучение структуры базы знаний, особенностей ее создания и на ручной ввод всех отношений, правил и параметров. Необходимо будет только подготовить знания. То есть представить их в определенном формате и загрузить в систему для создания готовой базы знаний.

В данной работе исследуется структура миварной базы знаний и определяется формат и структура входных данных для системы, которая на основе этих данных создаст файл базы знаний.

Описание структуры базы знаний

База знаний представляет собой совокупность параметров правил и отношений.

Класс – абстрактная сущность, обобщающее понятие. Класс может содержать в себе параметры и другие классы. Класс также имеет название, уровень иерархии и описание. В любой модели должен существовать хотя бы один класс. Класс с высшим уровнем иерархии называют корневым.[2]

Параметр — это объект, содержащий значение определенного типа: числового или текстового. Как и класс, параметр имеет название, уровень иерархии и описание. Кроме того, параметр может принимать какое-то значение по умолчанию.

Отношение — вид связи, использующий абстрактные переменные, описывающий их взаимодействие.

Правило — вид связи, привязывающий отношение к конкретным объектам.

Существует четыре типа отношения:

- 1) формула — математическая формула, состоящая из переменных и операторов;
- 2) условное отношение — продукционное отношение вида «если..., то..., иначе...»;
- 3) ограничение — продукционное отношение, накладывающее ограничение на значение параметров;
- 4) сложное отношение — программируемое отношение (код на языке JavaScript) для опытного пользователя.

Существуют следующие правила создания отношений:

1. Отношение записывается так, чтобы в левой части от знака «равно» стояли выходные переменные, а справа входные.

2. В отношениях можно использовать любые латинские буквы и символы операторов.
3. Переменным типа «строка» значения присваиваются следующим образом: $y = \text{“значение строки”}$.
4. Переменные в различных отношениях могут быть одинаковыми и абстрактными.
5. В отношении не могут быть одновременно одни и те же входные и выходные переменные.

База знаний представлена в виде файла формата xml. Ниже представлен пример базы знаний, содержащий три класса, шесть параметров, два отношения, два правила. Структура классов и параметров выглядит следующим образом:

```

Model 1
  Class 1
    Param 1
    Param 2
    Param 6
  Class 2
    Param 5
  Class 3
    Param 3
    Param 4

```

База знаний включает в себя совокупность классов, параметров, правил и отношений заданной предметной области и представлена в виде xml файла.

Ниже представлен пример базы знаний:

```

<model id="{5e6df39e-07d1-4a24-a0a7-1ac6eef79c83}" shortName="Model 1"
formatXmlVersion="2.0" description="Model 1">
<class id="{eef00543-d244-40e4-9303-609b2068fd92}" shortName="Model 1">
<parameters/>
<rules>
<rule id="c98cdb6e-10ae-41a0-8986-ee842a114f91" shortName="Rule 2" relation="eb0df91c-
926c-48cd-9c9a-df7abe2e6851" resultId="a:0d999107-6e94-474e-a86d-b88c99767956"
initId="d:04183e16-1ce8-4a30-9fa1-0f8ad53be14c;b:8f9f3353-e488-4e61-8d01-
cd5a4c9746e9;c:fe54f856-6db2-4af0-9fad-79ee52738582"/>
</rules>

```

```

<constraints/>
<classes>
<class id="1a24c361-19d0-4abf-b683-640057645548" shortName="Class 2">
<parameters>
<parameter id="0d999107-6e94-474e-a86d-b88c99767956" shortName="Param 5"
type="double"/>
</parameters>
<rules/>
<constraints/>
<classes>
<class id="65973b7f-1ab9-4e58-8346-ddf039568fe0" shortName="Class 3">
<parameters>
<parameter id="8f9f3353-e488-4e61-8d01-cd5a4c9746e9" shortName="Param 3" type="double"/>
<parameter id="fe54f856-6db2-4af0-9fad-79ee52738582" shortName="Param 4" type="double"/>
</parameters>
<rules/>
<constraints/>
<classes/>
</class>
</classes>
</class>
<class id="c43dc3d6-f670-49fe-9d6a-b15dfcc5a148" shortName="Class 1">
<parameters>
<parameter id="04183e16-1ce8-4a30-9fa1-0f8ad53be14c" shortName="Param 6" type="double"/>
<parameter id="2e2f4869-5f32-487f-9e5b-fec8203f0bfc" shortName="Param 1" type="double"/>
<parameter id="c5ad4453-fa5d-4a48-b782-7de860ff6de9" shortName="Param 2" type="double"/>
</parameters>
<rules>
<rule id="96c4cee0-9e6f-47c5-8a0d-452943180a95" shortName="Rule 1" relation="f37d07ef-
b071-43df-95fd-991911e8d8cf" resultId="a:04183e16-1ce8-4a30-9fa1-0f8ad53be14c"
initId="b:2e2f4869-5f32-487f-9e5b-fec8203f0bfc;c:c5ad4453-fa5d-4a48-b782-7de860ff6de9"/>
</rules>
<constraints/>
<classes/>
</class>

```

```

</classes>
</class>
<relations>
<relation id="eb0df91c-926c-48cd-9c9a-df7abe2e6851" shortName="Relation 2"
inObj="d:double;b:double;c:double" relationType="simple"
outObj="a:double">a=(b+c)/d</relation>
<relation id="f37d07ef-b071-43df-95fd-991911e8d8cf" shortName="Relation 1"
inObj="b:double;c:double" relationType="simple" outObj="a:double">a=b+c</relation>
</relations>
</model>

```

Корневой элемент в данной структуре файла (<model>) имеет два основных вложенных элемента – <class> и <relations>.

Каждый элемент <class> содержит вложенные элементы <parameters>, <rules>, <constraints> и <classes> и имеет следующие атрибуты:

- id – идентификатор класса
- shortName – название класса
- description – описание класса.

Элемент <rules> содержит правила <rule>, которые используют параметры данного класса. Если правило использует параметры классов разного уровня, то оно будет прописано в элементе <class> высшего уровня вложенности. Например, правило Rule 2 использует параметры классов Class 1, Class 2 и Class 3, и оно не может быть отнесено к одному из них. Поэтому элемент <rule>, описывающий данное правило, содержится в описании правил <rules> класса Model 1, который содержит в себе и Class 1, и Class 2, и Class 3.

Элемент <rule> содержит следующие атрибуты:

- id – идентификатор правила,
- shortName – название правила,
- relation – идентификатор отношения, к которому относится правило,
- resultId – идентификатор выходного параметра,
- initId – идентификаторы входных параметров,
- description – описание правила.

Элемент `<constraints>` включает ограничения `<constraint>` для параметров. `<constraint>` имеет следующие атрибуты:

- `id` – идентификатор ограничения,
- `shortName` – название ограничения,
- `relation` – идентификатор отношения, к которому относится ограничение,
- `initId` – идентификаторы входных параметров,
- `description` – описание ограничения.

Элемент `<parameters>` содержит параметры `<parameter>`, которые относятся к данному классу. Атрибуты элемента `<parameters>`:

- `id` – идентификатор параметра,
- `shortName` – название параметра,
- `defaultValue` – значение по умолчанию,
- `type` – тип параметра (`double`, `text`),
- `description` – описание параметра.

Элемент `<relations>` включает все отношения `<relation>` имеющиеся в базе знаний. Само отношение прописывается внутри элемента. Например:

```
<relation id="eb0df91c-926c-48cd-9c9a-df7abe2e6851" shortName="Relation 2"
inObj="d:double;b:double;c:double" relationType="simple"
outObj="a:double">a=(b+c)/d</relation>
```

Элемент `<relations>` содержит следующие атрибуты:

- `id` – идентификатор отношения,
- `shortName` – название отношения,
- `inObj` – типы входных параметров,
- `relationType` – тип отношения (`simple`, `constr`, `ifclause`, `prog`),
- `outObj` – тип входного параметра,
- `description` – описание отношения.

Определение формата и структуры входных данных

Для автоматизированного создания базы знаний необходимо определить формат и структуру входных данных, по которым она будет создаваться.

Входные данные можно представить в виде текстового файла или в формате `xlsx`.

Обработка текстового файла для выявления в нем параметров, правил и отношений сложнее, так как при создании текстового файла эксперт может допустить опечатки (лишний пробел, использование не того символа, неправильный перенос строки и т.д.). Предусмотреть все возможные варианты подобных ошибок и обеспечить их правильную обработку невозможно.

Поэтому лучшим вариантом формата входных данных является `xlsx`. Данные в табличном виде лучше поддаются структурированию.

Входные данные должны содержать в себе все параметры, классы, правила и отношения. Выделим пять таблиц для хранения этих данных. Первая таблица будет содержать параметры модели, вторая таблица – классы, третья – отношения и четвертая – правила, пятая – ограничения.

Таблица 1 представляет структуру для хранения данных о классах модели.

Таблица 1. Классы

id	shortName	description	parentClass
1	Model 1		-
2	Class 1		1
3	Class 2		1
4	Class 3		3

Столбцы `id`, `shortName` и `description` содержат данные соответствующих атрибутов элементов `<class>`. Поле `parentClass` служит для определения родительского класса, и позволит задать структуру классов модели.

В таблице 2 представлена структура данных, описывающих параметры базы знаний.

Таблица 2. Параметры

id	shorName	defaultValue	description	class
1	Param 1			2
2	Param 2			2
3	Param 3			4
4	Param 4			4
5	Param 5			3
6	Param 6			2

Столбцы id, shorName, defaultValue и description содержат данные соответствующих атрибутов элементов <parameter>. Поле class служит для определения класса, к которому относится параметр.

Так как система предназначена для автоматизации создания базы знаний на основе наборов формул, все параметры будут иметь числовой (double) тип данных. Следовательно, нет необходимости хранить информацию о типе параметров во входных данных, а также о типе входных и выходных параметров в отношениях.

В таблице 3 содержит структуру данных для отношений базы знаний.

Таблица 3. Отношения

id	shorName	inObj	relationType	outObj	description	relation
1	Relation 1	b;c	simple	a		$a=b+c$
2	Relation 2	b;c;d	simple	a		$a=(b+c)/d$
3	Relation 3	a	constr	flag		$a>10$

Столбцы id, shorName, inObj, outObj и description содержат данные соответствующих атрибутов элементов <relation>. Поле relation содержит само отношение (формулу).

Таблица 4 представляет структуру для хранения данных о правилах модели.

Таблица 4. Правила

id	shorName	relation	resultId	initId	description
1	Rule 1	1	6	1;2	
2	Rule 2	2	5	3;4;6	

Столбцы id, shortName, relation, resultId, initId и description содержат данные соответствующих атрибутов элементов <rule>.

В таблице 5 представлена структура данных, описывающих ограничения базы знаний.

Таблица 5. Ограничения

id	shorName	relation	initId	description
1	Constr 1	3	2	

Столбцы id, shortName, relation, initId и description содержат данные соответствующих атрибутов элементов <constraint>.

Данные таблицы будут храниться в одном файле формата xlsx. Каждая таблица будет находиться на отдельном листе. Следовательно итоговый файл должен иметь 5 листов со следующими названиями:

1. Classes
2. Parameters
3. Relations
4. Rules
5. Constraints

Программные средства для обработки данных

Прежде чем приступить к обработке `xlsx`-файла необходимо подготовить его в соответствии с некоторыми основными принципами, такими как:

- Первая строка таблицы обычно зарезервирована для заголовка, а первый столбец используется для идентификации единицы выборки;
- Избегать имен, значений или полей с пробелами. В противном случае каждое слово будет интерпретироваться как отдельная переменная, что приведет к ошибкам, связанным с количеством элементов на строку в наборе данных. По возможности, использовать:
 - подчеркивания,
 - тире,
 - разный регистр, где первая буква каждого слова пишется с большой буквы
 - объединяющие слова
- Короткие имена предпочтительнее длинных имен;
- стараться не использовать имена, которые содержат символы: `?, $, %, ^, &, *, (,), -, #, ? ,,, <, >, /, |, \, [,], {, и }`;
- Удалите все комментарии в файле, чтобы избежать добавления в ваш файл лишних столбцов или NA;
- Все пропущенные значения в наборе данных обозначить как NA.

Для создания программы будет использован язык программирования `python`. Для работы с `xlsx`-файлом, подготовленным экспертом, будут необходимы следующие библиотеки:

- `Pandas`
- `ElementTree`
- `lxml`

Pandas основана на NumPy и предоставляет простые в использовании структуры данных и инструменты анализа данных Python.

Данная библиотека позволяет прочитать файл с несколькими листами, с помощью метода `read_excel()`, если указать название листа в параметре `sheet_name`:

```
df = pd.read_excel('file_path/File.xlsx', sheet_name='List_1')
```

Также можно собрать словарь из датафреймов, в которых будут располагаться все таблицы файла. Чтобы получить нужный датафрейм, нужно обратиться к словарю по ключу с соответствующим названием листа:

```
excel_reader = pd.ExcelFile('file_path/File.xlsx')
sheet_to_df_map = { }

for sheet_name in excel_reader.sheet_names:
    sheet_to_df_map[sheet_name] = excel_reader.parse(sheet_name)
```

Далее с помощью функций и методов библиотеки Pandas будет производиться обработка датафреймов для последующего формирования базы знаний в формате xml.

ElementTree предоставляет простой способ создания XML-документов и записи их в файлы. Этой цели служит метод `ElementTree.write()`.

После создания объектом `Element` можно манипулировать, напрямую изменяя его поля (например, `Element.text`), добавляя и изменяя атрибуты (метод `Element.set()`), а также добавляя новых дочерних объектов (например, с помощью `Element.append()`).

Основы работы с `lxml` аналогичны `xml.etree.ElementTree`. Однако, библиотека `lxml` также предоставляет дополнительные возможности, такие как поддержка XPath и валидация схемы.

Заключение

В ходе данной работы была изучена структура миварной базы знаний. На основе полученной информации были определены формат и структура входных данных, а также были определены программные средства для дальнейшей обработки данных и создания базы знаний.

Список источников

1. Варламов О.О., Чувиков Д.А. Миварные технологии как средство создания систем автоматизации разумной деятельности человека // Автоматизация и управление в технических системах. 2016. № 1 (18). С. 13. EDN: ZXUHAT.
2. Варламов О.О., Чибирова М.О., Хадиев А.М., Антонов П.Д., Сергушин Г.С., Шошев И.А., Назаров К.В. Практикум по созданию миварных экспертных систем. Учебное пособие / под ред. О.О. Варламова. — М.: Изд-во НИИ МИВАР, 2016. — 184 с.