

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ДЛЯ
ЭКСПЕРТНОЙ СИСТЕМЫ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные
технологии
факультета КНиИТ
Кобзыревой Марины Владимировны

Научный руководитель:

к. ф.-м. н., доцент

А. С. Иванов

Зав. кафедрой:

к. ф.-м. н., доцент

С. В. Миронов

Саратов 2022

ВВЕДЕНИЕ

В настоящее время эффективность большинства программных продуктов характеризуется не только их функциональностью, но и удобством использования их возможностей. По этой причине пользовательский интерфейс, который обеспечивает обмен информацией между пользователем и программой, является неотъемлемым компонентом широкого ряда компьютерных систем. К их числу относятся и экспертные системы, которые применяются в решении множества узкоспециализированных задач и, следовательно, должны быть удобны в использовании. Главные задачи интерфейса в этом случае представляют собой интерпретацию запросов пользователя к базе знаний системы и формирование ответа в понятной пользователю форме.

В контексте взаимодействия с экспертной системой под пользователем, как правило, подразумевается человек, которому необходима консультация; однако в случаях, когда базу знаний необходимо обновить, в роли пользователя может выступать специалист. С целью упрощения обоих процессов работы было создано определённое количество как платных, так и бесплатных оболочек, имеющих разные характеристики и реализующих разные методы представления знаний. Именно эти программные средства выступают объектом исследования в данной работе.

Актуальность этого исследования, в свою очередь, обусловлена тем, что не все оболочки экспертных систем отличаются достаточным удобством для пользователя. Некоторые из них, в силу отсутствия графического интерфейса, поддерживают лишь взаимодействие через терминал, в то время как многие из существующих на данный момент приложений, объединяющих методы разработки экспертных систем с методами построения пользовательского интерфейса, находятся на экспериментальной стадии.

Целью бакалаврской работы является разработка интерфейса, поддерживающего возможности:

- создания базы знаний с нуля;
- открытия и редактирования уже существующей базы знаний;

— проведения консультации и получения рекомендации;
— предоставления пользователю объяснений касательно того, как был получен результат.

Поставленная цель определила **следующие задачи:**

1. определение понятия и описание структуры экспертной системы;
2. исследование роли интерфейса как компонента экспертной системы;
3. создание оконного приложения, отвечающего сформулированным выше условиям.

Методологические основы разработки экспертных систем представлены в работах К. Нейлора [1], Э. В. Попова [2], Л. Н. Ясницкого [3], Т. А. Гавриловой [4], Д. Джарратано [6], Д. В. Гаскарова [7].

Практическая значимость бакалаврской работы. В ходе выполнения бакалаврской работы было разработано приложение, представляющее собой оболочку для экспертной системы. Достоинством данной разработки является упрощение процесса конструирования баз знаний для экспертных систем.

Структура и объём работы. Бакалаврская работа состоит из введения, трёх разделов, заключения, списка использованных источников и двух приложений. Общий объем работы – 42 страницы, из них 34 страницы – основное содержание, включая 22 рисунка, цифровой носитель в качестве приложения, список использованных источников информации – 25 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Экспертные системы и их особенности» посвящён определению понятия экспертных систем, рассмотрению их структуры и классификации.

Под экспертной системой понимается программный комплекс, который объединяет возможности компьютера с экспертными знаниями и опытом в некоторой предметной области в такой форме, что система может предложить разумный совет или осуществить разумное решение поставленной задачи. Кроме того, желаемой характеристикой такой системы является её способность пояснять ход своих рассуждений в доступной пользователю форме. [1]

Основные принципы конструирования экспертных систем могут быть сформулированы в виде трёх положений.

1. Мощность системы обусловлена в первую очередь мощностью базы знаний и возможностью её пополнения и только во вторую очередь – методами, которые она использует.

2. Знания, позволяющие эксперту или экспертной системе получать качественные и эффективные решения задач, являются в основном эвристическими, экспериментальными, неопределёнными, правдоподобными. Необходимо подчеркнуть, что знания экспертов имеют индивидуальный характер, то есть свойственны конкретному человеку.

3. Учитывая неформализованность решаемых задач и эвристический, личностный характер используемых знаний, как пользователь, так и эксперт должны иметь возможность непосредственного взаимодействия с экспертной системой в форме диалога. [2]

Из изложенных выше представлений вытекают требования к архитектуре экспертной системы. Выделяют следующие компоненты:

1) база знаний – совокупность знаний предметной области, записанная на машинном носителе в понятной пользователю (эксперту) форме и представляющая собой ядро экспертной системы;

2) интеллектуальный редактор базы знаний – программа, представляющая возможность вносить изменения в базу знаний в диалоговом режиме;

3) интерфейс пользователя – комплекс программ, реализующих диалог пользователя с экспертной системой на стадии как ввода информации, так и получения результатов;

4) решатель (дедуктивная машина, блок логического вывода) – программа, моделирующая ход рассуждений эксперта на основании знаний, имеющихся в базе;

5) подсистема объяснений – программа, позволяющая пользователю получать ответы на вопросы о том, как была получена та или иная рекомендация и почему система приняла такое решение. [3]

Класс экспертных систем объединяет несколько тысяч различных программных комплексов, которые могут быть классифицированы по различным критериям. Наиболее часто экспертные системы различают по задаче, по связи с реальным временем, по степени интеграции, по предметной области, по методам представления знаний и по сложности. [4]

Второй раздел «Роль и элементы пользовательского интерфейса в создании экспертных систем» посвящён рассмотрению интерфейса как компонента экспертной системы.

Осуществляя работу с экспертной системой, специалист использует интерфейс для ввода информации и команд в систему и получения из неё выходной информации. Команды включают в себя параметры, которые направляют процесс обработки знаний, а информация обычно выдается в форме значений, присваиваемых определённым переменным.

Специалист может использовать четыре метода ввода информации: меню, команды, естественный язык и собственный интерфейс.

В качестве выходной информации технология экспертных систем предусматривает возможность получать не только решение, но и необходимые объяснения. Их обычно подразделяют на два типа:

— объяснения, выдаваемые по запросам (пользователь в любой момент может потребовать от экспертной системы объяснения своих действий);

— объяснения полученного решения проблемы (после получения решения пользователь может потребовать объяснений относительно того, как оно было получено. Система должна пояснить каждый шаг рассуждений, которые привели её к решению задачи).

Учитывая тот факт, что технологию работы с экспертной системой нельзя назвать простой, пользовательский интерфейс таких систем должен быть дружелюбным и не вызывать трудностей при ведении диалога. [5]

Кроме вышеперечисленного, в разделе были также освещены проблемы разработки пользовательских интерфейсов и рассмотрены некоторые существующие оболочки экспертных систем.

Третий раздел «Программная реализация» посвящён разработке приложения «ESysRedactor», представляющего собой оболочку для создания экспертной системы, а также демонстрации примера работы программы. Разработка приложения осуществлялась на языке Python с использованием библиотеки Tkinter, содержащей компоненты графического интерфейса пользователя. В качестве основного подхода к созданию программы были выбраны методы объектно-ориентированного программирования.

Первым делом в ходе разработки были реализованы классы «IfWorked», «IfTrue» и «VarType»: первый отвечает за переменные, сигнализирующие о том, сработало ли правило в процессе консультации; второй – за переменные, обозначающие истинность факта; третий определяет тип переменной (запрашиваемая, выводимая или запрашиваемо-выводимая).

Затем был создан класс «Variable», позволяющий создавать переменную в качестве элемента, используемого экспертной системой. Атрибутами объекта данного класса являются имя, домен, вопрос, задаваемый пользователю в ходе консультации, комментарий, служащий для удобства при создании системы, и тип (объект класса «VarType»). Данный класс имеет несколько методов, необходимых для получения или изменения значений атрибутов.

Далее был создан класс «Fact», отвечающий за формирование того или иного факта. Создаваемый объект имеет атрибуты переменной (объект класса «Variable»), её значения и истинности факта. Для упрощения работы были также реализованы несколько методов, среди которых большое значение для интерфейса имеет функция перевода объекта в строковый формат «Переменная = Значение».

Следующим был реализован класс «Rule», позволяющий сформировать продукционное правило. Его атрибуты – имя, переменная, сигнализирующая о срабатывании правила (объект класса «IfWorked»), список условий, комментарий-объяснение и вывод (объект класса «Fact»). Как и в случае с предыдущим классом, одним из методов, играющих ключевое значение, является функция перевода объекта в строковый формат «ЕСЛИ Факт, ТО Вывод».

Помимо вышеперечисленных классов был создан класс «DomainValues», необходимый для создания списков значений в том или ином домене.

Так как подразумевается, что в данном приложении настройка экспертной системы начинается с добавления доменов, первым было создано окно «Домены значений», также реализованное в форме класса. Среди его атрибутов можно выделить «self.domains» и «self.variables», представляющие собой упорядоченные словари и отражающие такие же атрибуты класса самой системы, о котором речь пойдёт немного позднее. Главными виджетами окна являются два элемента класса «Listbox» из библиотеки Tkinter, один для доменов и один для их значений. Для списка доменов был прописан обработчик событий, реагирующий на переключение между элементами списка. Список значений должен обновляться в зависимости от того, какой домен выбран в данный момент.

Далее было создано всплывающее окно «Редактор переменных». Как и в случае с окном «Домены значений», одним из основных атрибутов его класса был «self.variables», позволяющий создавать временный словарь, хранящий переменные и позволяющий не затрагивать одноимённый атрибут самой

системы до сохранения изменений. В этом окне тоже присутствуют элементы класса `Listbox` с прописанным обработчиком события выбора элемента из списка переменных, а также элемент класса `Combobox`, представляющий собой выпадающий список и позволяющий быстро менять домен переменной, с обработчиком события, моментально сохраняющим изменения и выводящим список соответствующих значений.

Последним из окон, играющих роль в создании экспертной системы, было создано окно «Правила». Основным атрибутом его класса является словарь «`self.rules`», временно хранящий правила. Окно содержит три элемента класса `Listbox`, а также поддерживает возможность добавления нового правила, изменения имени правила, удаления правила, добавления и удаления условий и вывода.

Основное окно содержит файловое меню, позволяющее открыть и сохранить файл, кнопки, вызывающие перечисленные выше вспомогательные окна, а также поддерживает возможность установки цели консультации, запуска самой консультации и предоставления объяснения. Наибольшую важность здесь имеет класс «`ExpertSystem`», определённые методы которого представляют собой механизм вывода, а именно – функция «`self.consult`», возвращающая установленный факт, и функция «`self.process_rule`», работающая с правилами и возвращающая для каждого значение истинности, ложности или неопределённости. Таким образом, принцип работы механизма вывода заключается в нахождении правил, в выводе которых фигурирует переменная, выбранная целью, а затем – в проверке каждого из правил до тех пор, пока не будет найдено сработавшее.

При наличии среды разработки, специалисты способны конструировать различные системы, работая только над созданием базы знаний. В различных случаях может возникнуть необходимость внесения ряда изменений в работу механизма логического вывода, однако в целом среда может быть использована для обработки знаний из любой предметной области. Как программа подобного рода, создаваемое в ходе данной практики приложение способно в некоторой

мере упростить разработку несложных экспертных систем и может оказаться инструментом, полезным для использования, к примеру, в учебных целях.

ЗАКЛЮЧЕНИЕ

В настоящей работе были проанализированы структура экспертной системы, её компоненты, типы классификации и режимы работы, а также сложности, которые связаны с разработкой интерфейса и, в частности, оболочек экспертных систем. На основании всего, что было рассмотрено выше, можно сделать вывод о важности роли пользовательского интерфейса в работе экспертных систем.

Кроме того, было разработано приложение, позволяющее пользователю создавать и редактировать небольшие поверхностные экспертные системы, проводить консультации и получать соответствующие результатам объяснения.

Таким образом, задачи, поставленные перед выполнением данной работы, были выполнены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 *Нейлор, К.* Как построить свою экспертную систему: Пер. с англ. / К. Нейлор. — Москва: Энергоатомиздат, 1991. — 286 с.
- 2 *Попов, Э. В.* Экспертные системы: Решение неформализованных задач в диалоге с ЭВМ / Э. В. Попов. — Москва: Наука, 1987. — 288 с.
- 3 *Ясницкий, Л. Н.* Введение в искусственный интеллект : учеб. пособие для студ. высш. учеб. заведений / Л. Н. Ясницкий. — Москва: Академия, 2010. — 176 с.
- 4 *Гаврилова, Т. А.* Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский. — Санкт-Петербург: Питер, 2000. — 384 с.
- 5 Технология и основные этапы построения интегрированных (корпоративных) и экспертных информационных систем [Электронный ресурс]. — URL: <http://expert-sistem.narod.ru/es1.html> (Дата обращения 28.05.2022). Загл. с экр. Яз. рус.
- 6 *Джарратано, Д.* Экспертные системы: принципы разработки и программирование, 4-е издание. : Пер. с англ. / Д. Джарратано, Г. Райли. — Москва: Вильямс, 2007. — 1152 с.
- 7 *Гаскаров, Д. В.* Интеллектуальные информационные системы. Учеб. Для вузов / Д. В. Гаскаров. — Москва: Высш. шк., 2003. — 431 с.