

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ

**РАЗРАБОТКА КОМБИНИРОВАННЫХ МЕТОДОВ ИЗВЛЕЧЕНИЯ
ЗНАНИЙ И ИХ ПРОГРАММНАЯ РЕАЛИЗАЦИЯ**

Выпускная квалификационная работа
обучающегося по направлению подготовки
09.03.03 «Прикладная информатика»
очной формы обучения, группы 07001404
Курнова Владислава Юрьевича

Научный руководитель:
к.т.н., доцент Зайцева Т.В.

БЕЛГОРОД 2018

СОДЕРЖАНИЕ

| | |
|--|----|
| ВВЕДЕНИЕ | 3 |
| 1 Анализ предметной области | 6 |
| 1.1 Проблемы приобретения и извлечения знаний и основные направления их решения | 6 |
| 1.2 Основные характеристики процессов извлечения экспертных знаний | 9 |
| 1.2.1 Участники процесса извлечения знаний, их функции и режимы взаимодействия | 10 |
| 1.2.2 Психологическая корректность процедуры извлечения знаний | 15 |
| 1.2.3 Полнота и непротиворечивость знаний | 17 |
| 1.2.4 Методы извлечения знаний | 18 |
| 1.3 Структурно-функциональная диаграмма процесса извлечения знаний «Как есть» | 20 |
| 1.4 Сравнительный анализ систем поддержки извлечения и приобретения знаний | 23 |
| 2 Проектирование информационной системы извлечения знаний комбинированным методом | 26 |
| 2.1 Структурно-функциональная диаграмма процесса извлечения знаний «Как должно быть» | 26 |
| 2.2 Определение вероятных комбинаций последовательностей методов извлечения знаний | 28 |
| 2.3 Алгоритм извлечения знаний комбинированным методом | 31 |
| 2.4 Модель базы данных для информационной системы извлечения знаний | 33 |
| 3 Программная реализация предлагаемого решения и определение его эффективности | 36 |
| 3.1 Разработка демонстрационного прототипа информационной системы извлечения знаний | 36 |
| 3.2 Контрольный пример реализации проекта | 44 |
| 3.3 Анализ социально-экономической эффективности проекта | 50 |
| 3.3.1 Определение эффективности предлагаемого подхода | 50 |
| 3.3.2 Расчет затрат на разработку информационной системы | 52 |
| ЗАКЛЮЧЕНИЕ | 58 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 60 |
| ПРИЛОЖЕНИЕ | 65 |

ВВЕДЕНИЕ

В последние годы внимание многих исследователей привлекают проблемы построения компьютерных систем, основанных на человеческих знаниях (СОЗ). Многие из этих проблем сами по себе не являются новыми — им уделяется большое внимание в рамках таких направлений исследований, как распознавание образов [15], принятие решений [19], решение проблем [44].

Новый всплеск интереса к этим проблемам связан, безусловно, с появлением и все более широким распространением таких разновидностей СОЗ, как экспертные системы [24, 27, 41] и системы поддержки принятия решений [19].

При построении СОЗ одной из центральных является задача передачи знаний эксперта компьютерной программе, известная также как проблема приобретения знаний. В первую очередь речь идет о личностных знаниях, об интуиции, «искусстве» решения проблем. Такие знания принято называть знаниями 2-го рода в отличие от объективных устоявшихся знаний, вошедших в учебники и справочники [8-10].

Знания 2-го рода (или, лучше сказать, умения) обычно передаются от учителя к ученику путем совместного решения задач, анализа удач и ошибок. Очевидно, что вопрос о передаче компьютеру знаний 2-го рода возникает при решении так называемых слабоструктуризованных проблем [45]. В этих проблемах основную роль играют качественные, трудноформализуемые, недостаточно определенные факторы. Человеческая интуиция, умения являются основой решения таких проблем.

Вопрос о выявлении и передаче компьютеру человеческих знаний и умений является крайне сложным. На пути решения этой задачи стоит множество препятствий. Упомянем основные из них, связанные с извлечением знаний, наименее разработанной в настоящее время частью проблемы приобретения знаний.

1) Человек не может сообщить общих абстрактных правил, которыми он руководствуется, решая ту или иную конкретную задачу. Человеческие умения чаще всего хранятся на подсознательном уровне.

2) В любой области человеческой деятельности имеется большое количество (десятки и сотни тысяч) возможных практических ситуаций принятия решений, при анализе которых проявляются умения человека. Желательно уметь передавать все эти возможности ЭВМ; что требует огромного труда и времени экспертов.

3) Люди, передающие компьютеру в том или ином виде свои знания и умения, неизбежно ошибаются. Чем бы ни была вызвана конкретная ошибка: усталостью, невнимательностью, трудностью ситуации, безошибочных экспертов, к сожалению не бывает.

Этап извлечения знаний касается выбора формы взаимодействия инженера по знаниям с экспертом. Основной вопрос, который должен быть решен, - это "как" взаимодействовать с экспертом, чтобы получить наиболее полное представление о предметной области и существующих в ней способах принятия решения.

Объектом исследования является процесс извлечения знаний у эксперта при его взаимодействии с когнитологом.

Предметом исследования является совокупность комбинированных методов извлечения знаний

Цель – сокращение времени работы экспертной группы за счет использования комбинированного метода извлечения знаний.

Задачи:

- проанализировать существующие методы извлечения знаний;
- проанализировать возможные подходы и способы извлечения знаний;
- построить модели традиционного и предлагаемого процессов извлечения знаний и проанализировать их;
- провести практическую апробацию и проанализировать полученные результаты.

Пояснительная записка к ВКР состоит из введения, трех разделов, заключения, списка используемых библиографических источников и приложений.

Во введении обосновывается выбор темы ВКР, рассматривается ее актуальность, выбираются объект и предмет исследования, формулируются цель ВКР и задачи, необходимые для ее достижения, описывается структура пояснительной записки.

В первом разделе рассматриваются теоретические вопросы, связанные с процессами приобретения и извлечения знаний, основные характеристики процесса извлечения экспертных знаний, методы и программные средства извлечения знаний, строится типовая модель извлечения знаний «как есть».

Во втором разделе рассматривается предлагаемый комбинированный метод извлечения знаний, строится модель процесса извлечения экспертных знаний «как должно быть», рассматриваются наиболее часто применяемые последовательности методов при извлечении знаний.

В третьем разделе описывается программная реализация предлагаемого комбинированного метода, приводится контрольный пример работы программы.

В заключении приводятся основные результаты ВКР и выводы по работе.

Список библиографических источников содержит актуальную литературу по рассматриваемым вопросам.

В приложении приведен фрагмент кода разработанного программного средства.

Выпускная квалификационная работа написана на 82 страницах, в ней присутствует 30 рисунков, 9 таблиц, 5 формул, а также приложение с программным кодом.

1 Анализ предметной области

1.1 Проблемы приобретения и извлечения знаний и основные направления их решения

В настоящее время в области инженерии знаний наблюдается некоторая терминологическая путаница, затрагивающая, в частности, и соотношение понятий приобретение и извлечение знаний. В работе рассматривается следующая точка зрения.

В соответствии с [7, 8, 16] под приобретением знаний понимается процесс получения знаний из различных информационных источников (выявление знаний) и преобразование их в компьютерную программу. Таким образом, приобретение знаний отождествляется со всем циклом построения прикладной СОЗ и включает в себя различные виды аналитической, синтетической и рутинной деятельности: сбор информации, формулировку проблемы, выбор или разработку формализма представления знаний, проектирование, программирование, отладку, документирование и т. д.

При выявлении знаний могут использоваться различные информационные ресурсы: монографии, учебники, статьи, отчеты, руководства, архивные материалы (содержащие примеры решения задач), базы данных, наблюдения за функционированием реальных систем и, конечно же, люди — эксперты.

Под извлечением знаний понимаются процессы выявления знаний, необходимых для построения СОЗ, когда источником информации являются эксперты в проблемной области. Таким образом, понятия извлечение и приобретение знаний находятся в соотношении часть — целое, а понятия выявление и извлечение знаний — в соотношении общее — частное.

Особая сложность приобретения знаний заключается в том, что процесс построения прикладных СОЗ оказывается длительным, трудоемким и слабоструктурированным, причем существенная часть всех трудностей приходится па долю извлечения знаний.

Выделяют пять основных направлений исследований, ориентированных на решение проблемы приобретения знаний.

1) Разработка формализмов представления и манипулирования знаниями, языковых и программно-инструментальных средств построения соз;

2) Создание методов и систем автоматического и автоматизированного анализа естественно-языковых текстов.

3) Построение методов и программных систем машинного обучения.

4) Разработка и адаптация методов и программных систем поддержки извлечения экспертных знаний.

5) Создание методологии и, на ее основе, операциональной технологии приобретения знаний, обеспечивающей поддержку полного цикла разработки прикладных СОЗ.

Первое направление исследований является фундаментом, определяющим возможности современных СОЗ и отражает в первую очередь «компьютерные аспекты» их построения [9, 10, 34, 45]. В рамках этого направления наработан широкий спектр формализмов представления знаний [18, 28].

Второе и третье направления исследований представляют собой попытку обойти трудности, связанные с извлечением экспертных знаний. Методы и системы анализа текстов на естественном языке являются перспективным направлением исследований [24]. Так же развитые системы обработки текстов находят все более широкое применение в качестве составной части систем приобретения знаний, обеспечивая автоматизированную обработку протоколов интервьюирования и вербальных отчетов.

Третье направление исследований заключается, главным образом, в разработке индуктивных алгоритмов, позволяющих преобразовывать знания

экспертов, зафиксированные в виде структурированных примеров решения практических задач, в базу знаний СОЗ. Исследования в области машинного обучения рассмотрены в [10, 26, 33, 43]. Некоторые из алгоритмов индуктивного обучения входят в состав коммерческих инструментальных систем [41, 43, 47].

Четвертое направление исследований отражает «человеческие аспекты» построения СОЗ и связано с разработкой и применением методов и программных средств поддержки извлечения знаний. Эти методы и системы предназначены для повышения продуктивности труда эксперта и инженера по знаниям, участвующих в разработке прикладной СОЗ, и, в конечном итоге, должны приводить к сокращению сроков разработки и улучшению качества базы знаний конечной системы. Важность этого направления обусловлена значимостью экспертов как источника информации, дефицитностью экспертизы и трудоемкостью процессов извлечения знаний.

Ценность экспертов как источника информации определяется следующими причинами. Во-первых, эксперты являются владельцами так называемых знаний второго рода, определяющих их «профессиональную культуру» [33] и состоящих из эмпирических правил, эвристик, интуитивных соображений, которые, главным образом, основаны на личном опыте и, как правило, не могут быть получены из других источников. Во многих случаях эти знания являются единственной основой хорошего решения практических задач.

Во-вторых, эксперты являются активным источником информации. Необходимая информация может эксплицироваться экспертом в концентрированном и организованном виде самостоятельно или в ответ на вопросы инженера-когнитолога. Кроме того, в процессе разработки системы иногда возникает эффект обратной связи, приводящий к более глубокому пониманию экспертом предметной области, пробелов в своих знаниях и, в результате, к самосовершенствованию. Тема извлечения знаний привлекает внимание учёных как в Европе, так и во всём мире [8, 11, 17, 18, 20, 23, 26, 27, 31, 33, 36, 40, 44].

Процессы извлечения знаний отличаются высокой трудоемкостью и большими интеллектуальными нагрузками, приходящимися на эксперта и инженера по знаниям, связанными с вербализацией экспертизы, управлением процессом коммуникации и необходимостью освоения, анализа и документирования больших объемов новой информации.

Как отмечается в [34], разумной стратегией исследований в данной области является разработка арсенала разнообразных методов и программных средств извлечения знаний, их систематизация и выработка рекомендаций о целесообразности использования соответствующих подходов в зависимости от реальных условий.

Такая стратегия обусловлена тем, что цели, функции и некоторые другие характеристики извлечения знаний могут существенно отличаться на разных этапах построения СОЗ. Кроме того, известно, что даже в рамках одной ограниченной области экспертизы существуют различные виды знаний, для извлечения которых могут оказаться полезными разные методы.

Пятое направление исследований ориентировано на создание промышленной методологии и технологии построения СОЗ, агрегирующей последние достижения и перспективы в области инженерии знаний. Это направление должно обеспечить переход с современного уровня знаний о том, как осуществляется разработка СОЗ, и характеризуемого как искусство, на уровень операциональной технологии [23].

1.2 Основные характеристики процессов извлечения экспертных знаний

Несмотря на то, что глобальной целью извлечения знаний является получение ответа на первый и основной вопрос, решаемый при построении СОЗ: «что представлять», локальные цели и некоторые другие характеристики

различных подходов и процедур извлечения знаний могут существенно отличаться.

К разряду таких характеристик можно отнести следующие: участники процесса извлечения знаний, их функции и режимы взаимодействия; этап разработки СОЗ, на котором выполняется извлечение знаний, объем и характер получаемой при этом информации; психологическая корректность процедуры извлечения знаний, надежность, полнота и непротиворечивость получаемой информации [9].

1.2.1 Участники процесса извлечения знаний, их функции и режимы взаимодействия

Извлечение знаний представляет собой коммуникационный процесс, в котором могут принимать участие эксперт, инженер по знаниям (инженер-когнитолог [24] или просто когнитолог) и программная система. В дальнейшем будем предполагать, что эксперт не является специалистом в области разработки СОЗ, а когнитолог не является экспертом в проблемной области. В действительности возможны ситуации, когда один и тот же человек одновременно совмещает качества эксперта и когнитолога, но это должно рассматриваться скорее как исключение из правила, а не типичный случай.

Эксперт может быть определен как человек, обладающий исключительными познаниями и практическими навыками (умениями) решения определенного круга проблем в рассматриваемой предметной области.

В литературе отмечаются следующие необходимые и желательные качества экспертов [10, 38]: эффективность, производительность, понимание, способность обучаться и самообучаться, трактовка противоречий, знание ограничений, способность к объяснениям.

Вербализация является основной функцией, выполняемой экспертом в процессе извлечения знаний, и может рассматриваться как отображение

внутреннего, когнитивного уровня знаний эксперта на внешний, лингвистический уровень в рамках следующей трехуровневой модели приобретения знаний.

На когнитивном уровне знания эксперта о проблемной области выражены на языке внутренних, когнитивных структур. На вопрос о том, как именно представлены знания на когнитивном уровне, как происходит их пополнение и как они «работают» при решении экспертом проблем, в настоящее время не существует удовлетворительных ответов, поскольку механизмы и закономерности работы человеческого мозга пока недостаточно изучены [18].

На следующем, лингвистическом уровне знания и умения эксперта представлены в виде знаковых структур, выраженных в рамках лингвистических примитивов его профессионального языка, и включают в себя представления как конкретных элементов реального мира, так и абстрактные понятия и обобщающие категории, описывающие классы сущностей, связей, процессов и т. д.

Если два первых уровня относились к человеку, то третий относится к компьютеру, т. е. к операционализации уже полученных от человека знаний. На этом операциональном уровне знания и умения эксперта, а также знания, полученные из других источников информации, выражаются в рамках некоторого выбранного или сформированного операционального способа представления, реализованного при помощи программных средств.

Выделяют три возможных роли, выполняемые экспертом при вербализации: эксперт как информатор; эксперт как решатель проблем; эксперт как учитель [26, 38]. В первом случае эксперт выполняет функции справочника — отвечает на поставленные вопросы. Вторая роль связана с решением экспертом реальных или модельных задач. Выступая в роли учителя, эксперт эксплицирует не только знания, но и их организацию.

Когнитолог — относительно новая специальность, появившаяся в результате естественного разделения труда разработчиков СОЗ (выделяют

также программистов и менеджеров проекта) и во многом похожая на деятельность системного аналитика (консультанта по принятию решений, постановщика задач) при разработке традиционных программных систем.

Можно выделить две наиболее существенные отличительные черты специальности когнитолога [9, 10]. Во-первых, когнитолог должен знать принципы и методы искусственного интеллекта, принятия решений и когнитивной психологии, применяемые при построении СОЗ. Это в первую очередь относится к знанию существующего арсенала методов и программных средств извлечения знаний, способов представления знаний, механизмов логического вывода и программно-инструментальных средств разработки СОЗ, хотя детальное знание последнего пункта в большей степени относится к компетенции программистов.

Вторая существенная отличительная черта профессии когнитолога связана с объемом и типом получаемой информации. Системного аналитика главным образом интересует постановка задачи, требования пользователей и ограничения, предъявляемые к разрабатываемой системе с тем, чтобы в последующем были созданы или подобраны подходящие алгоритмы решения выделенных задач и реализованы в виде традиционной программной системы.

Когнитолог, помимо такой «постановочной» информации, должен выявить разнообразные знания и умения эксперта, позволяющие ему решать идентифицированные проблемы. Следовательно, инженер-когнитолог должен обладать знаниями психолога, умением вести беседу с экспертом, знать возможности и ограничения человеческой системы переработки информации.

Полученная от эксперта информация должна быть систематизирована и документирована для того, чтобы программисты могли в дальнейшем использовать ее при реализации системы, решающей проблемы так же, как это делает эксперт.

Выделим следующие основные взаимосвязанные функции, выполняемые когнитологом при извлечении знаний: управление, переработка и хранение полученной информации.

Управление процессом извлечения знаний необходимо для выполнения целенаправленного отбора адекватной информации, т. е. для выявления только тех знаний, которые необходимы для построения СОЗ, и с такой степенью полноты и детальности, которые достаточны для обеспечения необходимого качества работы системы. Управление реализуется в форме последовательности содержательных сообщений (вопросов, заданий), поступающих к эксперту от когнитолога в процессе их коммуникации.

Функция переработки включает в себя всевозможные способы и процедуры анализа поступающей от эксперта и синтеза новой информации, среди которых необходимо отметить следующие: идентификация и конструирование терминов и понятий; выяснение и фиксация их смыслового содержания; выявление и установление различных видов отношений между лексическими элементами, например таких, как синонимия, омонимия, обобщение/конкретизация, часть/целое и т.д.; различные виды проверок информации на полноту и непротиворечивость; формирование моделей экспертизы и т.д.

Функция хранения заключается в запоминании, выборке и документировании полученной от эксперта и синтезированной информации.

Программные средства, рассматриваемые как партнер, участвующий в процессе извлечения знаний, в настоящее время не могут полностью заменить когнитолога на всем цикле разработки и используются для поддержки его деятельности, позволяя в различной степени автоматизировать функции хранения и анализа информации и, реже, функцию управления.

Программные средства, используемые при извлечении знаний, могут быть разбиты на три категории: вспомогательные программы, системы извлечения знаний и системы приобретения знаний. К группе вспомогательных программ, используемых при извлечении экспертных знаний, могут быть отнесены различные текстовые процессоры, системы управления базами данных, программы анализа данных, обеспечивающие различные виды переработки и хранения информации.

Системы извлечения знаний характеризуются тем, что их основной функцией является управление процессом извлечения знаний. Системы приобретения знаний обеспечивают выполнение более широкого набора функций. Их обобщенная архитектура включает: блок извлечения знаний; средства автоматизированного анализа текстов; обучающие алгоритмы; системы представления и управления данными и знаниями; подсистемы выбора операционального формализма представления знаний и инструментальных средств разработки; блок компиляции и преобразования информации в различные представления; подсистему генерации отчетов и документации.

Можно выделить три возможных режима взаимодействия участников процесса извлечения знаний, отражающих степень его автоматизации: ручной, полуавтоматизированный и полностью автоматизированный режимы. В ручном режиме функции управления, переработки и хранения информации полностью выполняет когнитолог.

В полуавтоматизированном режиме некоторые из этих функций выполняет программная система, пользователем которой, как правило, является когнитолог. В третьем, полностью автоматизированном режиме все функции когнитолога в течение некоторого периода времени выполняет программная система, которая ведет непосредственный диалог с экспертом. В этом случае предъявляются повышенные требования к организации дружественного человеко-машинного интерфейса.

В реальных ситуациях при разработке конкретных СОЗ извлечение знаний в разные моменты времени может выполняться в разных режимах.

При этом отмечается определенная закономерность, связывающая этапы разработки СОЗ, на которых выполняется извлечение знаний, и доминирующий режим взаимодействия участников работы. В настоящее время на ранних этапах разработки преобладает рутинный режим, на заключительном этапе полностью автоматизированный режим оказывается предпочтительней.

1.2.2 Психологическая корректность процедуры извлечения знаний

Режимы и стадии извлечения знаний являются как бы внешними характеристиками процесса извлечения знаний. Важнейшей внутренней характеристикой процедуры извлечения знаний является ее психологическая корректность.

При любом методе извлечения знаний источником информации является человек. Следовательно, необходимо принимать во внимание основные характеристики человеческой системы переработки информации.

Достоверность информации существенно зависит от двух факторов: понятность и привычность операций, выполняемых экспертом при извлечении знаний, и соответствие операций по переработке информации возможностям человеческой системы.

В последние годы получено немало результатов психологических исследований, позволяющих предъявить определенные требования к способам получения информации от человека. Так, данные психологических исследований показывают, что многие достаточно распространенные способы получения информации (назначение весов признаков, назначение вероятностей и т. д.) являются некорректными: они ведут к противоречиям, смещениям, ошибкам [19]. Анализ показывает, что количество психологически корректных операций получения информации в широко распространенных на практике многоаспектных задачах сравнительно невелико. Не так просто извлекать знания, используя лишь корректные способы получения информации от человека, но только эти способы ведут к надежным базам знаний.

Вторая проблема состоит в определении допустимой нагрузки на эксперта. Эта нагрузка определяется рядом параметров, характеризующих рассматриваемую задачу. В качестве примера возьмем задачу многоаспектной классификации. В этой задаче эксперт рассматривает объект, характеризующийся рядом аспектов, и относит его к одному из нескольких

классов решений. Так, например, врач относит сочетание симптомов, характеризующих пациента, к одному или нескольким возможным заболеваниям.

Если классы решений упорядочены (например, заболевания упорядочены по степени тяжести), то возникает задача порядковой классификации. Поведение человека при решении этой задачи исследовалось в серии психологических экспериментов [19]. Результаты экспериментов показывают, что люди ведут себя достаточно надежно и непротиворечиво при определенных значениях таких параметров, как количество аспектов, число возможных значений для каждого из них и количество классов решений. Если значения этих параметров увеличиваются и переходят определенную границу, то поведение людей резко меняется. Часто при этом резко возрастает количество ошибок и противоречий.

Существуют определенные пределы человеческих возможностей для многих операций получения информации от человека. Эти пределы крайне важно знать при разработке методов извлечения экспертных знаний.

Результаты ряда психологических исследований позволяют следующим образом охарактеризовать систему хранения и переработки информации у эксперта. Прежде всего, хороший эксперт обладает знанием тысяч конкретных ситуаций, которые хранятся в его долговременной памяти. По оценке Г. Саймона, это число достигает 30—50 тысяч [44, 45].

Каждая из этих ситуаций описана на привычном для эксперта профессиональном языке. Умение гибко использовать эти ситуации при решении конкретной задачи, вероятнее всего, хранится на подсознательном уровне [41]. Все вспомогательные операции по извлечению и переработке информации совершаются в кратковременной памяти. Ее малый объем служит причиной использования человеком различных эвристических приемов. Эти приемы часто ведут к ошибкам и противоречиям.

1.2.3 Полнота и непротиворечивость знаний

Еще одной важной характеристикой, связанной с получением информации от экспертов, является полнота извлечения и представления знаний в СОЗ и непротиворечивость (безошибочность) базы знаний. Под полнотой и непротиворечивостью базы знаний в общем случае понимают способность системы формировать непротиворечивое решение для любой ситуации в рамках ограниченной проблемной области.

Рассмотрим источники неполноты и ошибок базы знаний СОЗ, используя введенную ранее трехуровневую модель приобретения знаний. Знания и умения эксперта на когнитивном уровне, отражающие его квалификацию и степень проработанности проблемной области, могут быть неполными и содержать ошибочные представления о реальности.

При извлечении знаний, интерпретируемом в рамках трехуровневой модели как отображение когнитивного уровня на лингвистический, может возникать два типа негативных эффектов, приводящих к появлению дополнительных ошибок и уменьшению полноты представления на лингвистическом уровне.

Во-первых, на лингвистическом уровне могут появляться ошибки, связанные как с неточностью и неоднозначностью естественного профессионального языка эксперта, так и с вербализацией неверных представлений эксперта.

Во-вторых, отображение адекватных знаний когнитивного уровня на лингвистический может быть неполным. Это происходит по двум причинам. С одной стороны, эксперт может не понимать, какие именно из его знаний необходимо эксплицировать, и, с другой стороны, не все знания эксперта могут быть адекватно вербализованы (например, знания из области дегустации вин, духов и т.д.).

И, наконец, при отображении знаний с лингвистического на операциональный уровень представления также может происходить уменьшение точности и полноты описания реальности. Основным источником этих искажений является несовершенство операциональных способов представления знаний.

В ряде методов извлечения знаний получаемая информация проверяется на непротиворечивость (один из способов выявления ошибок) и полноту. Так, например, в методе, реализованном в системе КЛАСС [19], проверки на непротиворечивость осуществляются в ходе извлечения знаний, причем при выявлении противоречий эксперту дается возможность их анализа и исправления. В данном случае проблема описывается на языке признаков и под полнотой понимается наличие в базе знаний решений для любых сочетаний значений этих признаков.

1.2.4 Методы извлечения знаний

Существуют различные классификации методов извлечения знаний [7-11]. Одна из классических классификаций методов извлечения знаний представлена на рисунке 1.1. Каждый метод имеет свои достоинства и недостатки, а также сферу применения.

На выбор метода извлечения знаний влияют 3 фактора:

- личностные особенности инженера знаний,
- личностные особенности эксперта,
- характер предметной области.



Рисунок 1.1 – Классификация методов извлечения знаний

Известный вывод о том, что по мере накопления опыта эксперт все больше утрачивает умение словесно выражать свои знания (парадоксальный факт Джонсона). Доказательствами выступают достаточно убедительные психологические результаты исследований, согласно которым люди часто не в состоянии свои мыслительные процессы достоверно описать. Следует также отметить теоретика искусственного интеллекта Марвина Минского, который писал «самосознание — это сложная, но тщательно сконструированная иллюзия...» и «.. только как исключение, а не как правило, человек может объяснить то, что он знает».

Таким образом, необходима процедура взаимодействия инженера по знаниям с источником знаний (экспертом), позволяющая сделать явными рассуждения специалистов при принятии решений и структуру их представлений о предметной области. Можно представить обобщенную схему извлечения знаний. Рассмотрим подобную обобщенную схему на примере коммуникативных методов.



Рисунок 1.2 – Обобщенная схема извлечения знаний коммуникативными методами

В случае применения текстологических методов знания экспертов рассматриваются в виде срезовых однопериодных знаний, а зоной ответственности когнитолога являются все действия, производимые в процессе извлечения знаний.

1.3 Структурно-функциональная диаграмма процесса извлечения знаний «Как есть»

На рисунке 1.3 представлена контекстная диаграмма процесса извлечения знаний традиционным на сегодняшний день способом. В качестве входящей информации рассматриваются как первичные, так и вторичные знания эксперта, выходом являются непосредственно извлеченные знания, управление – документы, регламентирующие процесс извлечения знаний, а

механизмами – когнитолог и эксперт / эксперты (в зависимости от количества одновременно привлекаемых специалистов).



Рисунок 1.3 – Контекстная диаграмма модели процесса «Как есть»

Так как, как правило, одного метода извлечения знаний бывает недостаточно, то был проведен анализ количества используемых методов когнитологом для каждой поставленной задачи и определено, что наиболее часто используются три метода (реже используется два метода и только в особых случаях использует четыре и больше методов, один метод при извлечении знаний используют только для построения демонстрационного прототипа СОЗ) [9-11].

На рисунке 1.4 представлена декомпозиция контекстной диаграммы, которая представлена в виде трех параллельных блоков: извлечение знаний у эксперта с помощью первого метода, извлечение знаний у эксперта с помощью второго метода, извлечение знаний у эксперта с помощью третьего метода. Методы выбираются когнитологом интуитивно. Часто, для удачного подбора используемых методов извлечения экспертных знаний, необходим серьезный опыт, у начинающих когнитологов бывают провальные подборы методов.

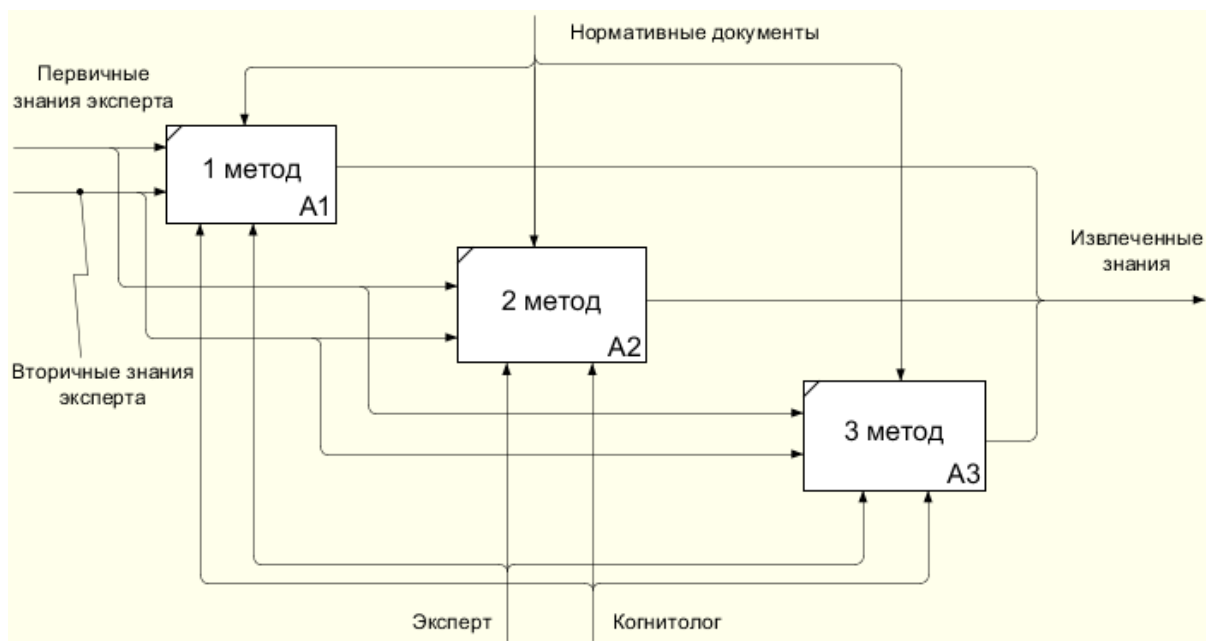


Рисунок 1.4 – Декомпозиция контекстной диаграммы модели процесса извлечения знаний «Как есть»

Анализ показал, что полученные знания после каждого метода являются в своем роде уникальными и редко согласуются со знаниями, полученными другими методами. Кроме того, при использовании данного подхода, после каждого метода меняется предметная и / или проблемная область. При таком подходе процесс извлечения знаний очень часто становится итерационным, когда извлеченные знания после каждого метода необходимо уточнять.

Кроме того, данный подход не позволяет проводить работу одновременно с несколькими экспертами при извлечении знаний всеми методами. А работа с каждым экспертом в индивидуальном порядке должна отвечать психологическим, лингвистическим и гносеологическим аспектам извлечения знаний в режиме реального времени [23, 24, 30].

Кроме того, следует учитывать, что при использовании текстологических методов вероятность неправильного понимания замысла эксперта у когнитолога достаточна велика.

Также когнитолог использует технические средства ограниченно только для оформления извлеченных знаний. Механизм «технические средства» на

данной диаграмме не представлен, так как появляется только на диаграммах декомпозиции 2-го уровня.

Таким образом, данный подход приводит к значительным как трудовым (и эксперта, и, особенно, когнитолога), так и к временным затратам, и очень часто финансовым.

1.4 Сравнительный анализ систем поддержки извлечения и приобретения знаний

Проблемы приобретения и извлечения знаний с помощью персональных компьютеров связаны с разработкой специальных информационных технологий, которые бы обеспечивали поддержку процедур извлечения и структурирования знаний. С первой созданной системы приобретения знаний по сегодняшний день можно выделить три стадии.

На первой стадии появилось первое поколение систем приобретения знаний на базе «оболочек» экспертных систем. Процессы извлечения и структурирования знаний выполнялись человеком. Подсистема приобретения знаний служила для ввода знаний в БЗ и ее корректировки. Экспертные системы заполнялись знаниями по следующей схеме:

- создание конкретной экспертной системы;
- опустошение базы знаний;
- разработка системы приобретения знаний для нового наполнения БЗ;
- формирование базы знаний для другой экспертной системы.

На второй стадии появились системы приобретения знаний второго поколения, основанные на предварительном детальном анализе предметной области и моделях, позволяющих рассматривать процедуры извлечения, структурирования и формализации знаний как процесс преобразования лингвистических знаний в другие представления и структуры. Существенное

влияние на системы второго поколения оказала психосемантика, на базе которой были созданы инструментальные средства многомерного шкалирования, факторного анализа, репертуарных решеток, логического вывода.

Третья стадия развития систем приобретения знаний связана с созданием автоматизированных средств приобретения знаний. При этом структура БЗ формируется в процессе приобретения знаний, а не заранее.

Множество существующих и потенциально возможных систем приобретения знаний можно отобразить классификацией (табл. 1.1).

Таблица 1.1 – Методы и системы приобретения знаний

| Метод приобретения знаний | Наименование системы и авторы | Характеристика |
|----------------------------|--|---|
| 1 | 2 | 3 |
| Структурированное интервью | RESIAS (Davis R.) ROGET (Bennet J.) SALT (Markus S.) MOLE (Eshelman L.) OPAL (Muzen M.) МЕДИКС (Ларичев О.И.) | Формирует новые понятия и правила Производит концептуальную организацию знаний для диагностических ЭС Формирует базы знаний в области конструирования методом пошагового распространения ограничений Обеспечивает контекстное приобретение знаний на основе структурированного интервью Обеспечивает формирование и наращивание БЗ экспертной системы, дающей советы по лечению онкологических больных Использует процедуры экспертной классификации для независимых свойств, признаков и их значений. Повышение эффективности экспертной классификации обеспечивается за счет применения априорно заданного отношения линейного порядка на множестве состояний |
| Имитация консультаций | АРИАДНА (Моргоев В.) ЭСКИЗ (Андриенко Г.) | Реализует метод многократного решения экспертом проблемы классификации в режиме последовательной вопросно-ответной консультации Включает набор игр для приобретения знаний, являющихся модификациями метода репертуарных решеток |

Продолжение таблицы 1.1.

| 1 | 2 | 3 |
|--|--|---|
| Интегрированные среды приобретения знаний | AQUINAS (Boose J.) KITTEEN (Shaw M.) | Содержит набор программных средств для извлечения экспертных знаний разными методами: средства анализа репертуарных решеток с последующим преобразованием системы конструкторов в базу продукционных правил; методы конструирования иерархических структур знаний; средства извлечения и представления неточных знаний; подсистемы тестирования, пополнения и коррекции базы знаний и др. Основана на построении и анализе репертуарных решеток. В отличие от AQUINAS данная интегрированная среда обеспечивает извлечение элементов из тестов, анализирует примеры решения задач экспертом и генерирует продукционные правила |
| Приобретение знаний из текстов | KRITON (Diderich J.) ТАКТ (Kaplan R.) | Выявляет процедурные знания на основе метода протокольного анализа из книг, документов, описаний, инструкций Выделяет из предварительно подготовленного текста объекты, процессы и отношения каузального характера |
| Инструментарий прямого приобретения знаний | SIMER + MIR (Осипов ГС.) | Позволяет формировать модели и базы знаний предметной области с неясной структурой объектов, неполно описанным множеством свойств объектов, большим набором разнородных связей между объектами |

В первом разделе рассмотрена предметная область: изучены вопросы приобретения и извлечения знаний и основные вопросы их развития, выделены основные характеристики процессов извлечения экспертных знаний, построена диаграмма процесса извлечения знаний (как есть), проведен анализ систем поддержки извлечения и приобретения знаний.

2 Проектирование информационной системы извлечения знаний комбинированным методом

2.1 Структурно-функциональная диаграмма процесса извлечения знаний «Как должно быть»

На рисунке 2.1 представлена контекстная диаграмма процесса извлечения экспертных знаний. От диаграммы «как есть» она отличается наличием третьего механизма – информационной системы извлечения знаний.

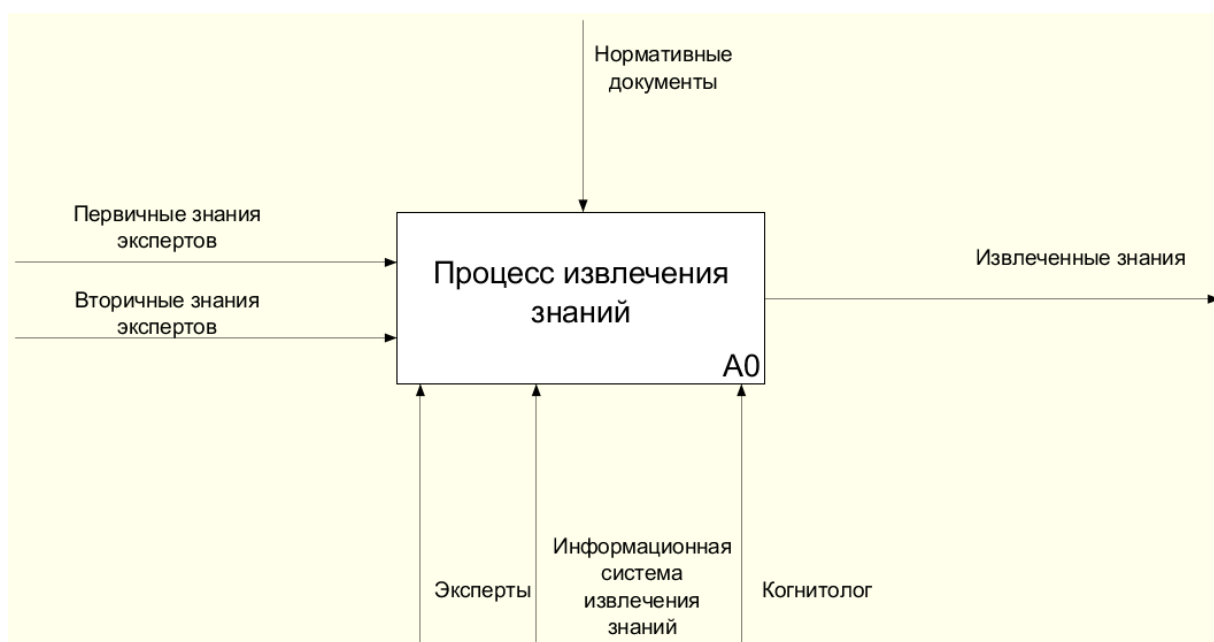


Рисунок 2.1 – Контекстная диаграмма процесса извлечения экспертных знаний «Как должно быть»

Декомпозиция контекстной диаграммы представлена на рисунке 2.2. На данном рисунке видно, что в отличие от традиционного способа, когда методы могли быть использованы как параллельно, так и последовательно в течении значительного промежутка времени, в предлагаемом методе все методы используются только последовательно, при этом после каждого метода

происходит, во-первых, формирование поля знаний, с которым в дальнейшем работают эксперты при извлечении знаний последующими методами, и, во-вторых, разработка инструкций, которые не позволяют выйти экспертам за рамки предметной и / или проблемной области.

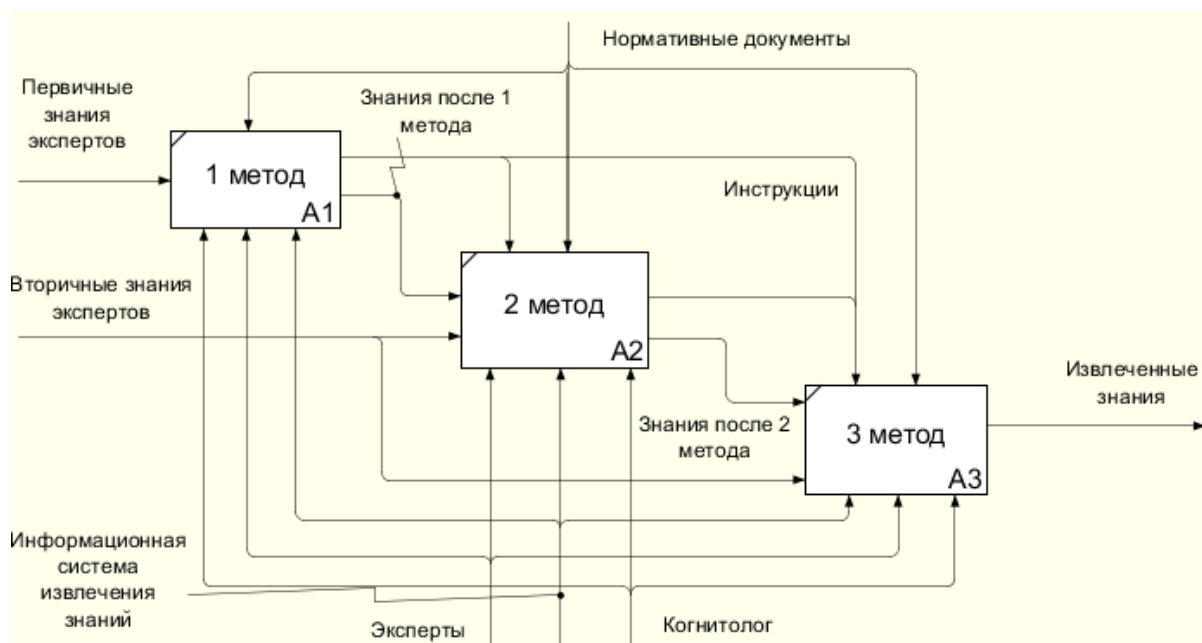


Рисунок 2.2 – Декомпозиция контекстной диаграммы процесса извлечения экспертных знаний «Как должно быть»

Данный подход позволяет:

- одновременно работать с несколькими экспертами, при необходимости уйти от персоналий и обозначать экспертов либо по номерам, либо по определенным именам;
- использовать то, что знания в процессе извлечения каждым последующим методом базируются на тех, которые были получены предыдущим;
- отвечать психологическим, лингвистическим и гносеологическим аспектам извлечения знаний в удобном для когнитолога режиме;
- обеспечивать при использовании текстологического метода как одного из трех консультацию с экспертами, то есть оценивать извлеченные знания специалистами в своей области, а не когнитологом.

2.2 Определение вероятных комбинаций последовательностей методов извлечения знаний

Используя литературные источники можно определить наиболее вероятные комбинации методов извлечения знаний [8-10, 32, 34, 35, 38]. Во-первых, необходимо определить наиболее оптимальное количество методов, используемых последовательно для извлечения экспертных знаний. Во-вторых, рассмотреть наиболее часто встречающиеся комбинации методов.

В первом разделе указывалось, что наиболее часто встречаются комбинации трех методов извлечения знаний, однако необходимо в этом убедиться, используя статистические данные. Представленная ниже информация была получена на основе открытых источников.

На рисунке 2.3 представлена диаграмма распределения количества методов, используемых при извлечении экспертных знаний. Как видно из рисунка в почти половине случаев используют 3 последовательно применяемых метода (46%), 2 метода используют в трети случаев.

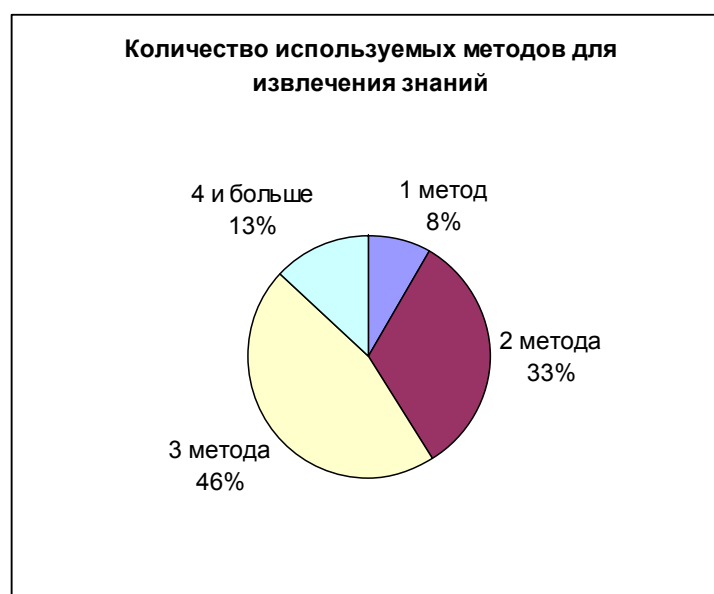


Рисунок 2.3 – Диаграмма распределения количества используемых методов извлечения знаний

Рассмотрим наиболее часто встречающиеся последовательности 2-х и 3-х методов извлечения экспертных знаний. На рисунке 2.4 представлены наиболее вероятные комбинации двух групп методов извлечения знаний. Все остальные комбинации в сумме не превышают 3%, поэтому на диаграмму вынесены не были.

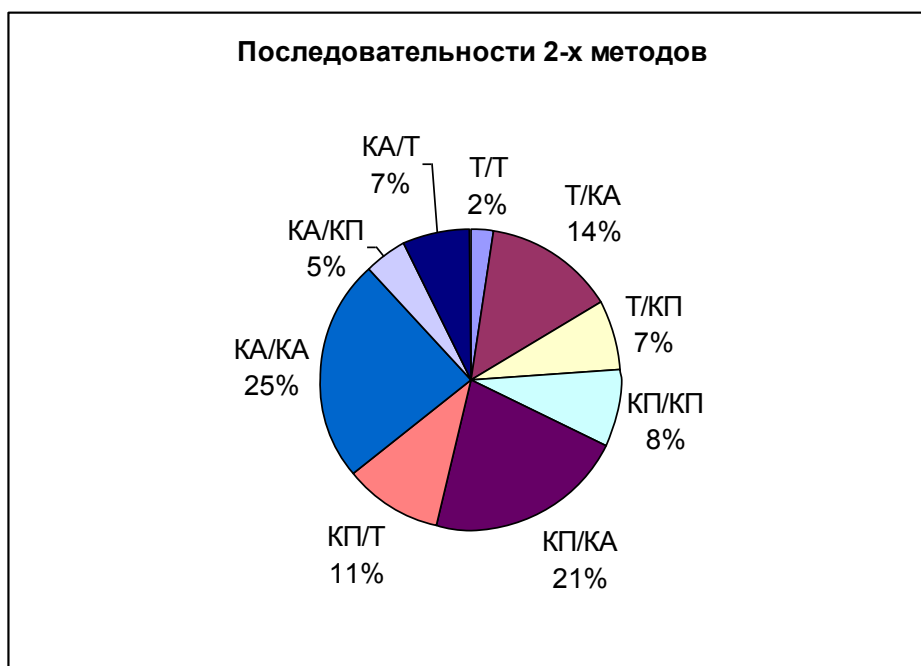


Рисунок 2.4 – Диаграмма распределения применяемых последовательностей двух групп методов извлечения экспертных знаний

На основе анализа библиографических источников были выделены наиболее часто встречающиеся последовательности 3 поочередно применяемых групп методов к решению конкретной задачи. Как видно, чаще всего выбирается следующая схема:

- один из текстологических методов / один из пассивных коммуникативных методов / один из активных индивидуальных коммуникативных методов или один из активных групповых коммуникативных методов (23%);

- в качестве первого метода используется один из пассивных коммуникативных методов, затем один из текстологических методов и один из

активных индивидуальных коммуникативных методов или один из активных групповых коммуникативных методов (18%);

- один из активных индивидуальных коммуникативных методов или один из активных групповых коммуникативных методов / один из текстологических методов / один из активных индивидуальных коммуникативных методов или один из активных групповых коммуникативных методов (11%);

- один из текстологических методов / один из активных индивидуальных коммуникативных методов или один из активных групповых коммуникативных методов / один из пассивных коммуникативных методов (10%).

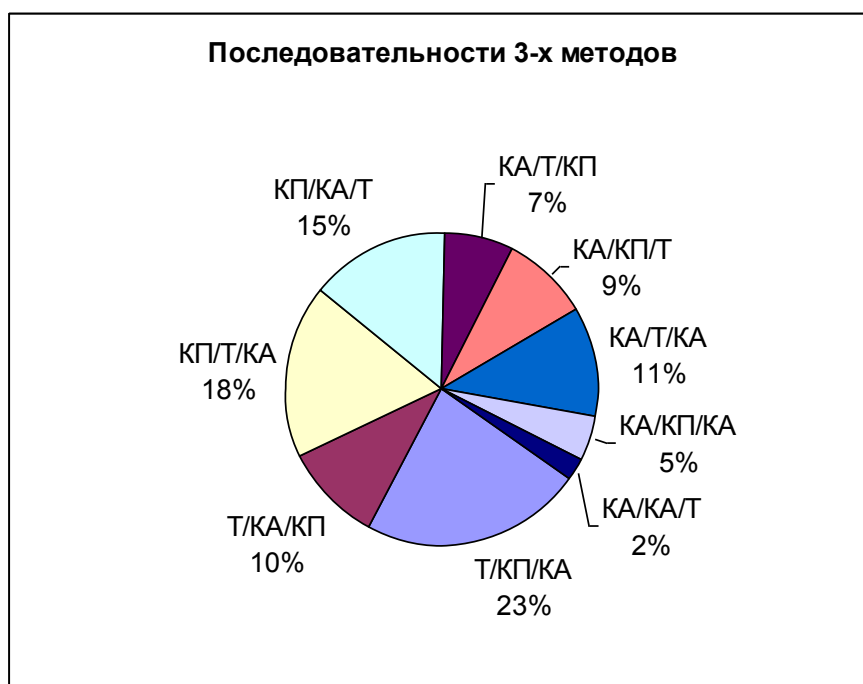


Рисунок 2.5 – Диаграмма распределения применяемых последовательностей трех групп методов извлечения экспертных знаний

Остальные последовательности в виду редкого использования в ВКР не рассматривались, хотя можно привести пример, в котором в качестве первых двух методов используются пассивные методы, а затем один из активных.

Рассмотрим последовательности конкретных методов извлечения знаний. В литературе встречаются следующие комбинации методов (рис. 2.6):

- один из текстологических методов / лекция / круглый стол;
- один из текстологических методов / лекция / мозговой штурм;
- лекция / протокол «мысли вслух» / один из методов ролевых игр;
- один из текстологических методов / лекция / диалог;
- наблюдение / мозговой штурм / один из текстологических методов;
- лекция / круглый стол / один из текстологических методов и другие.

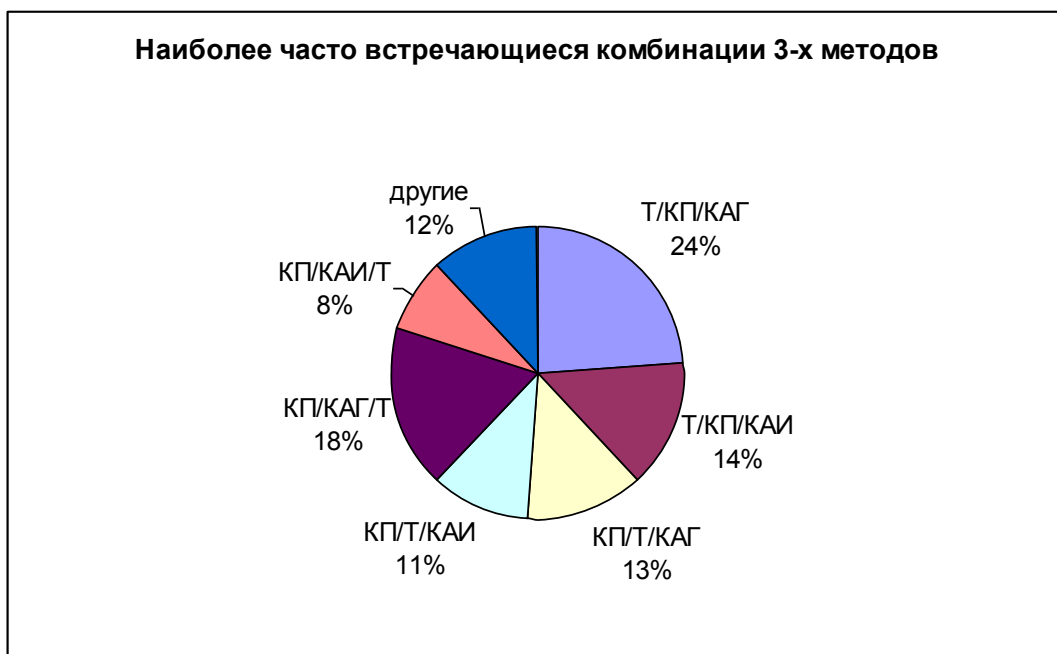


Рисунок 2.6 - Диаграмма распределения применяемых последовательностей трех методов извлечения экспертных знаний

2.3 Алгоритм извлечения знаний комбинированным методом

На рисунке 2.7 представлен алгоритм, по которому происходит извлечение знаний комбинированным способом. Сначала выбирается количество методов извлечения экспертных знаний, которые будут использоваться для решения выбранной задачи. Данное число выбирается исходя из опыта и интуиции когнитолога.

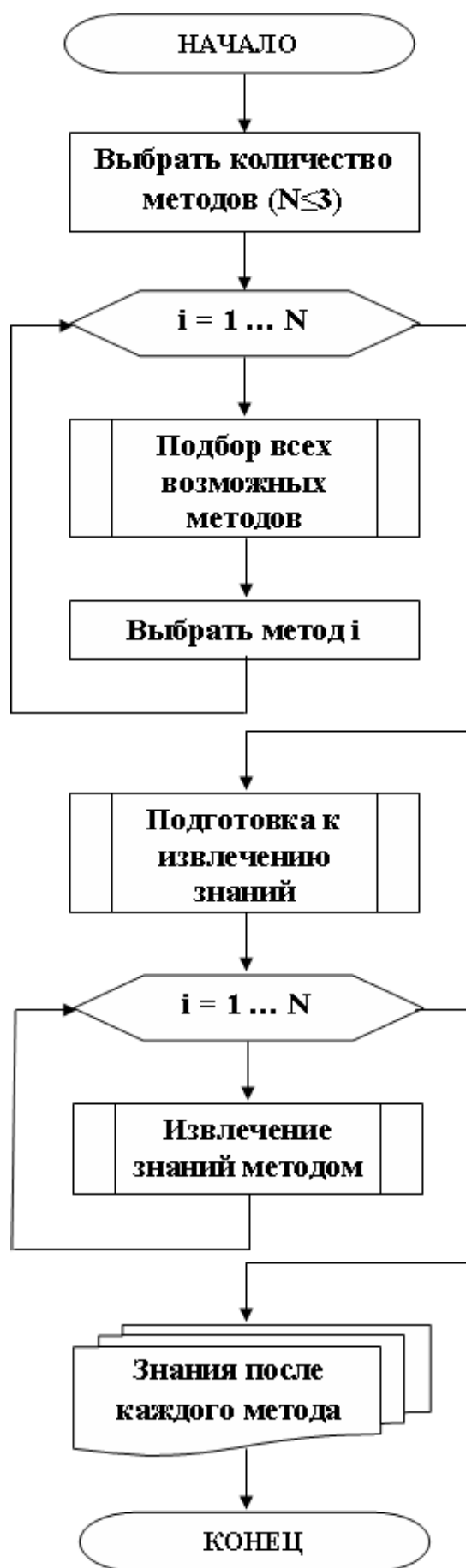


Рисунок 2.7 – Алгоритм последовательного извлечения знаний

Далее для каждого метода сначала осуществляется подбор всех возможных методов извлечения знаний, исходя из информации о предыдущем выбранном методе, а затем когнитолог из предложенного списка выбирает метод. Так при выборе первого метода извлечения экспертных знаний когнитологу для выбора предлагается полный перечень всех возможных методов, после его выбора происходит отбор только тех методов, которые могут применяться после ранее выбранного.

После выбора всех последовательно применяемых методов извлечения знаний происходит поэтапная процедура извлечения знаний, при этом на каждом последующем шаге учитывается информация, полученная на предыдущих.

После извлечения знаний последним выбранным методом формируются документы, в которых можно просмотреть: во-первых, итоговый результат процедуры, а во-вторых, знания, получаемые после каждого использованного метода.

2.4 Модель базы данных для информационной системы извлечения знаний

Для представления диаграммы схемы данных использовалось программное средство MS Visio. Диаграмма схемы базы данных представлена на рисунке 2.8. База данных состоит из 8 таблиц:

- процедура извлечения знаний (характеризует процесс извлечения экспертных знаний по конкретной задаче, включает предметную и проблемную область, когнитолога, выбранного для проведения процедуры, эксперта / экспертов, отобранных когнитологом согласно предметной области, выбранный метод / методы извлечения знаний, сроки проведения процедуры);

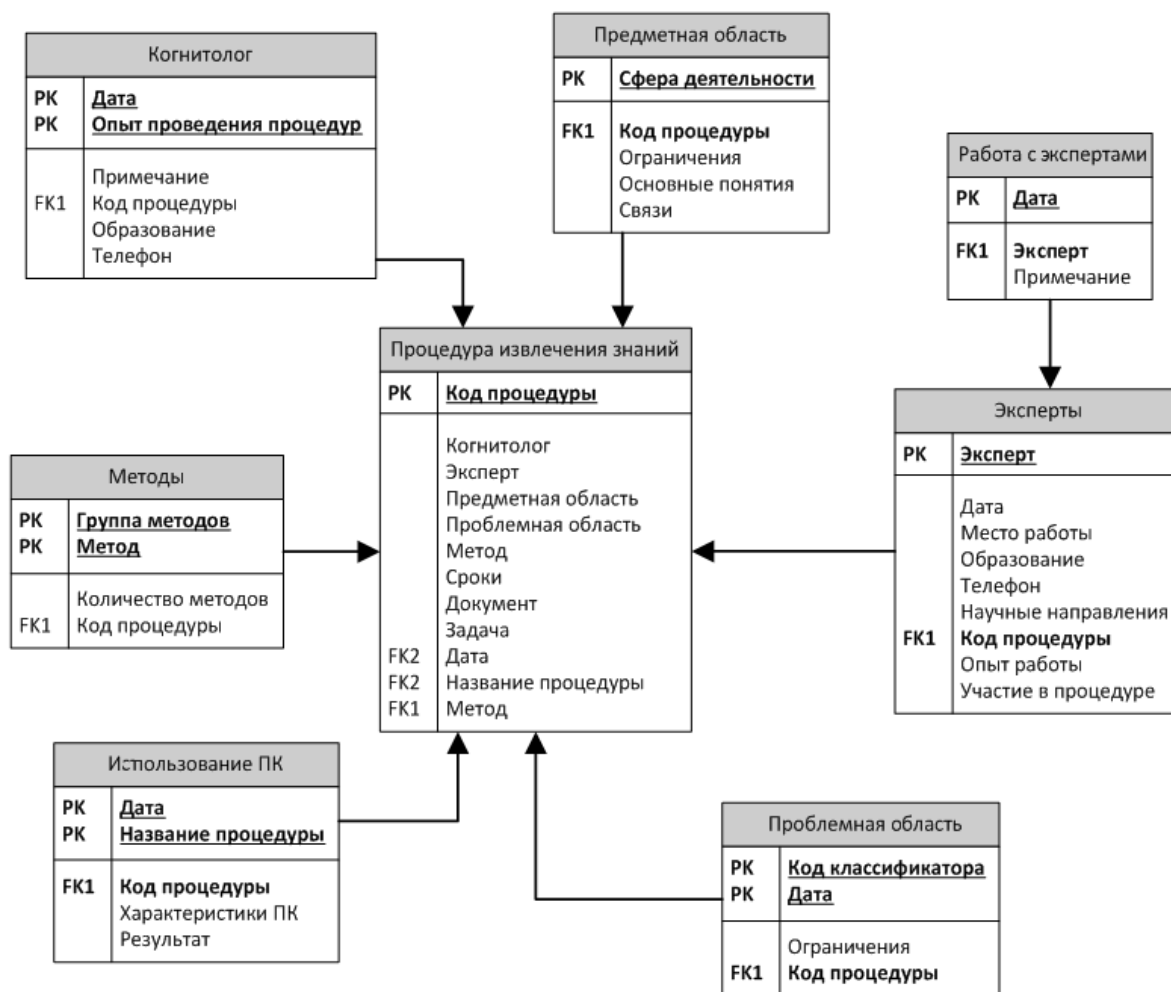


Рисунок 2.8 - Диаграмма схемы базы данных

- эксперты (содержит информацию обо всех экспертах, включая образование, научные направления работы, опыт работы и предыдущие участия в процедурах извлечения знаний);
- работа с экспертами (содержит информацию об отборе экспертов для решения задач);
- когнитолог (содержит информацию о когнитологах, которых можно привлечь к проведению процедуры извлечения экспертных знаний, их образовании и опыте работы в качестве когнитолога, дополнительную информацию);
- предметная область (содержит перечень предметных областей с указанием сфер применения и возможностью выбора ограничений, а также перечень основных понятий каждой области);

- проблемная область (возможность выбора проблемы из перечня, ее возможные ограничения);
- методы (включает разбиение методов по группам и определяет возможное количество используемых методов с предварительным отсечением);
- использование персонального компьютера (содержит аппаратно-программные характеристики ПК).

Во втором разделе рассмотрен процесс проектирования информационной системы извлечения знаний комбинированным методом: построена диаграмма процесса извлечения экспертных знаний (как должно быть), выявлены наиболее вероятные комбинации последовательно применимых методов извлечения знаний, описан алгоритм извлечения знаний комбинированным методом, построена модель базы данных для информационной системы извлечения знаний.

3 Программная реализация предлагаемого решения и определение его эффективности

3.1 Разработка демонстрационного прототипа информационной системы извлечения знаний

В качестве программной реализации метода прогнозирования данных был выбран программный продукт C# Builder и язык программирования C#.

C# Builder- программный продукт, инструмент быстрой разработки приложений (RAD), интегрированная среда программирования (IDE), система, используемая программистами для разработки программного обеспечения на языке C#.

Данный язык программирования был выбран потому, что большое количество разработанных интеллектуальных систем написаны либо на нем, либо на родственном.

Для проектирования базы данных использовалась кроссплатформенная реляционная система управления базами данных FirebirdSQL. Firebird является сервером баз данных. Один сервер Firebird может обрабатывать несколько сотен независимых баз данных, каждую с множеством пользовательских соединений. Среди недостатков Firebird можно выделить: отсутствие кэша результатов запросов, полнотекстовых индексов, значительное падение производительности при росте внутренней фрагментации базы, ограниченное количество одновременных подключений.

В данной работе рассматривается программная реализация демонстрационного прототипа информационной системы извлечения знаний с помощью комбинирования методов, поэтому рассматривается ограниченное число решаемых задач с уже определенными предметной и проблемной областями, аппаратно-программными характеристиками используемого оборудования и выбранным когнитологом. Таким образом, в разработанном

прототипе можно выбирать количество методов (от 1 до 3), непосредственно методы извлечения знаний и привлекаемых экспертов (от 1 до 4).

Реализованный прототип информационной системы состоит из двух модулей:

- модуль отбора методов извлечения экспертных знаний;
- модуль непосредственного извлечения знаний комбинированным способом.

В первом модуле реализован алгоритм выбора наиболее подходящего последующего метода извлечения знаний исходя из выбранного на предыдущем этапе метода.

Главная страница разработанной информационной системы представлена на рисунке 3.1.

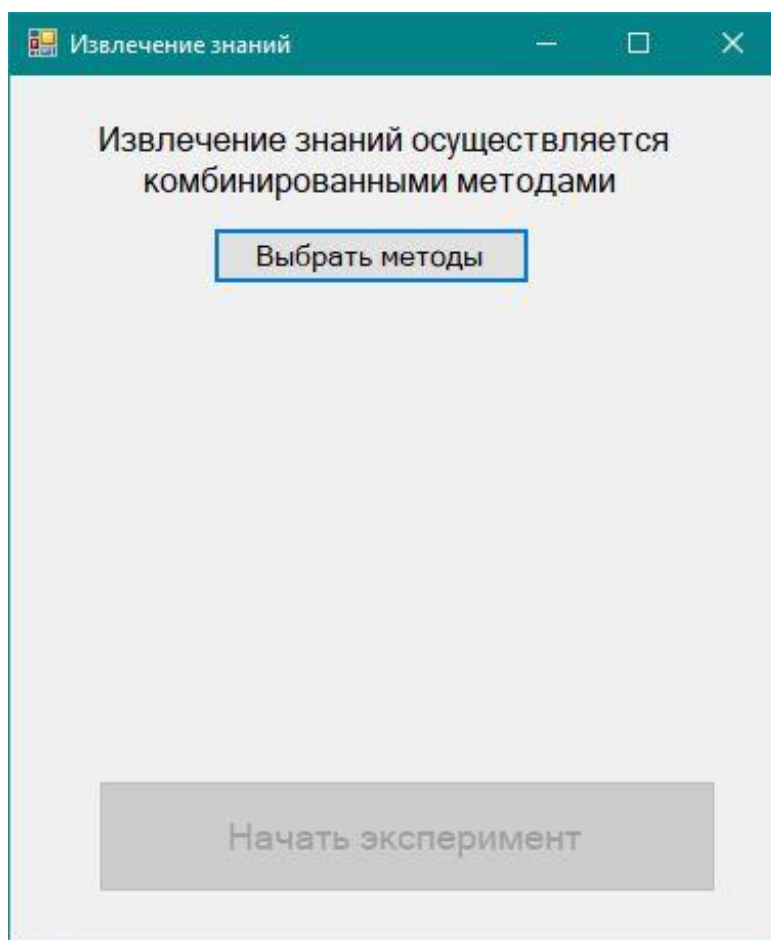


Рисунок 3.1 – Главная страница

При нажатии кнопки «Выбрать методы» пользователю (когнитологу) предлагается выбрать первый метод (рис. 3.2). При нажатии на активную область появляется выпадающий список всех методов извлечения экспертных знаний (рис. 3.3). Данный список сформирован исходя из выбранной классификации методов извлечения знаний в 1-ом разделе ВКР.

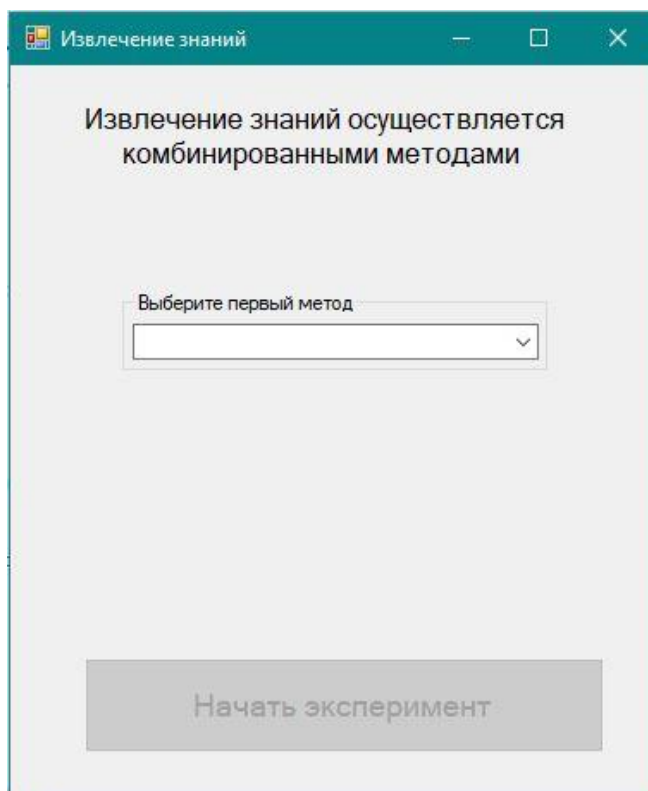


Рисунок 3.2 – Выбор 1-го метода

После выбранного 1-го метода когнитолог может или остановиться (тогда в процедуре извлечения знаний будет использоваться только один метод) или продолжить выбор методов. На рисунке 3.4 представлен выпадающий список методов извлечения знаний, из которых можно выбрать 2-ой используемый в процедуре метод. Как видно из рисунка предлагаемый список имеет перечень методов, отличный от первоначального. Это связано с тем, что после выбора первого метода не только происходит подбор возможных последующих методов, но и, используя статистические данные, список

располагается в виде уменьшения вероятности использования метода в зависимости от выбранного ранее.

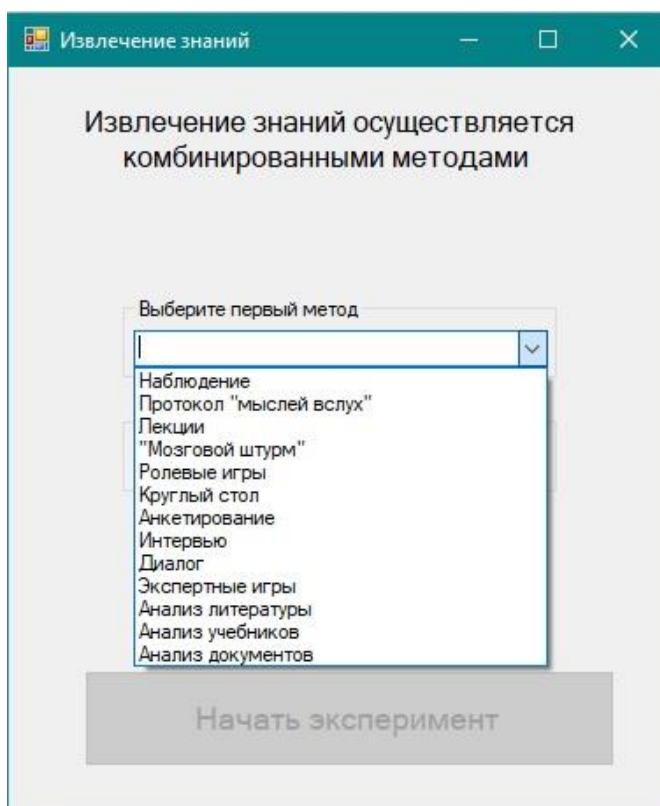


Рисунок 3.3 – Выпадающий список для выбора 1-го метода

После выбранного второго метода когнитолог может или остановиться и перейти непосредственно к организации процедуры извлечения знаний, или выбрать 3-й метод.

На рисунке 3.5 представлен процесс выбора третьего метода извлечения знаний. Как видно из рисунка в данном случае список предлагаемых методов еще уменьшился. Однако, в ряде случаев, в зависимости от выбранного 2-го метода список может измениться в любую сторону – увеличиться или уменьшиться, или остаться прежним (если выбран в качестве 2-го метода метод аналогичный 1-му).

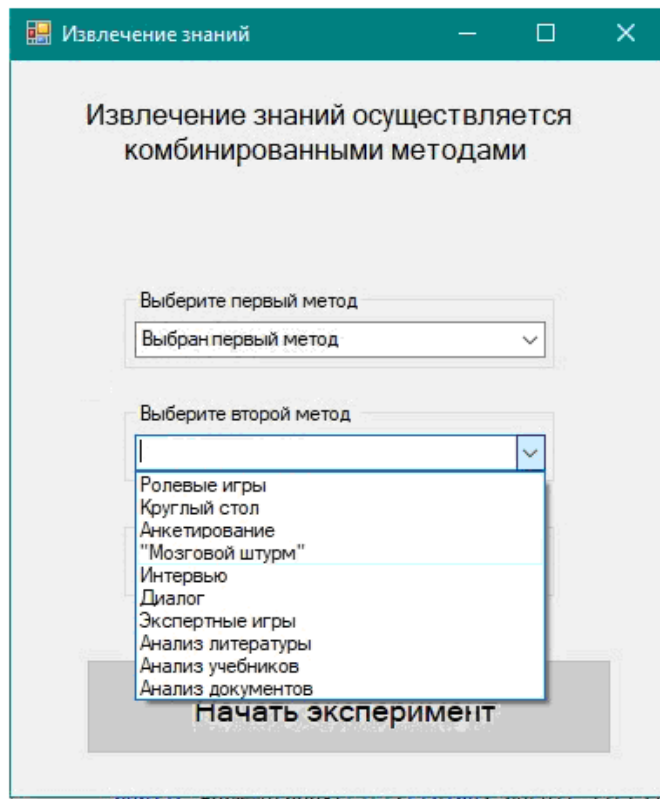


Рисунок 3.4 – Выбор 2-го метода извлечения знаний

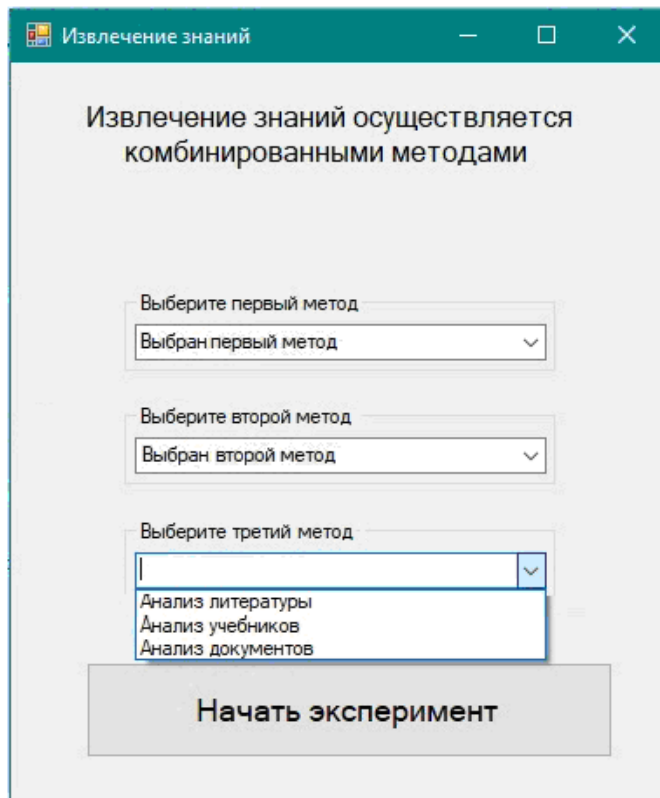


Рисунок 3.5 – Выбор 3-го метода извлечения знаний

На рисунке 3.6 представлено экранное окно программы, в котором осуществлен выбор всех методов. Следующим шагом является проведение процедуры извлечения знаний, для чего необходимо нажать кнопку «Начать эксперимент». При нажатии кнопки происходит переход ко второму модулю – непосредственному проведению процедуры извлечения экспертных знаний комбинированным способом.

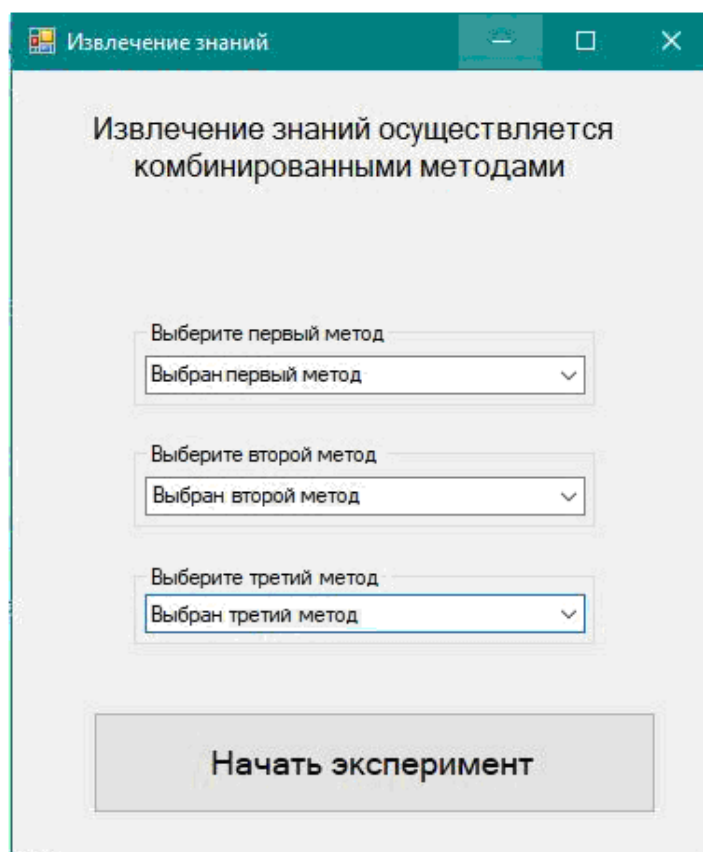


Рисунок 3.6 – Выбраны три последовательных метода извлечения знаний

В разработанном демонстрационном прототипе отсутствует подготовительная часть, связанная с предварительной работой по определению предметной и / или проблемной области выбранной задачи, отбором основных понятий и определений предметной области и связи между ними, а также отбором эксперта / экспертов для привлечения к решению задачи.

На рисунках 3.7-3.9 представлены экранные окна информационной системы, которые загружаются в зависимости от того из какой группы методов

извлечения знаний выбранный на данном этапе метод. На рисунке 3.7 представлено экранное окно при работе с группой пассивных коммуникационных методов, 3.8 – активных коммуникационных методов, 3.9 – текстологических методов.

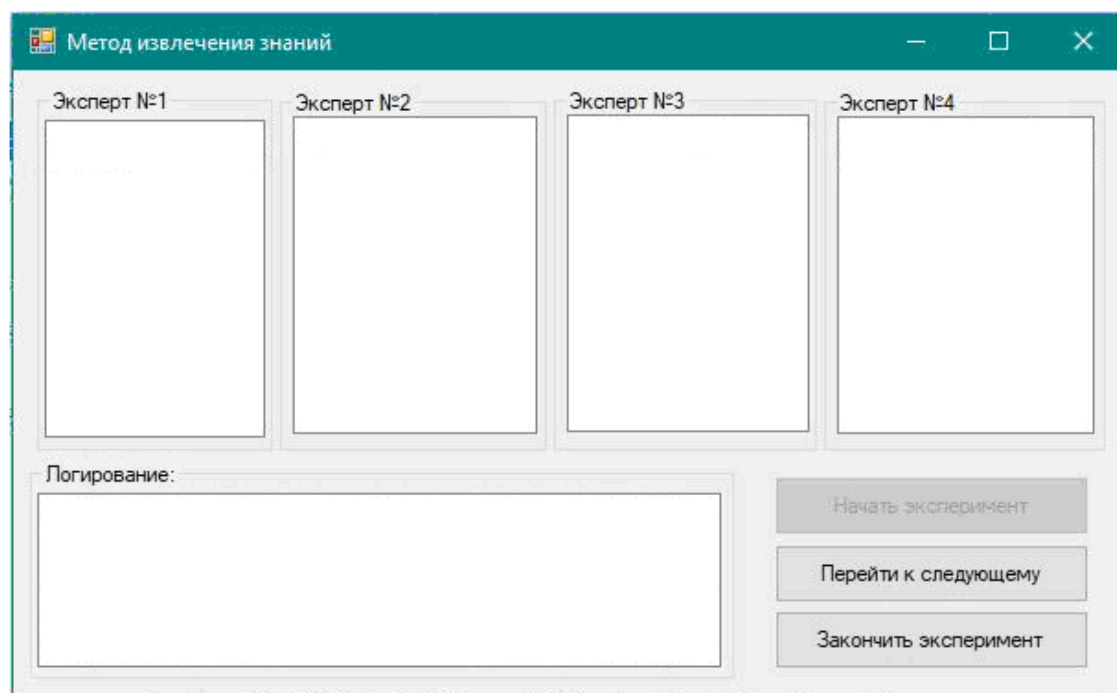


Рисунок 3.7 – Рабочее окно при выборе пассивного коммуникативного метода

В каждом окне присутствуют следующие элементы управления:

- для всех групп методов:

1) кнопка «Начать эксперимент» (начинает процедуру по извлечению экспертных знаний выбранным методом);

2) кнопка «Перейти к следующему» (позволяет перейти к следующему методу извлечения знаний, в том случае если метод является последним из выбранных, данная кнопка отсутствует);

3) кнопка «Закончить эксперимент» (позволяет закончить процедуру извлечения экспертных знаний либо по окончании эксперимента – были использованы для извлечения знаний все выбранные методы, либо в случае сбоя – технического, программного

или учитывающего человеческий фактор, либо в случае необходимости прервать процедуру);

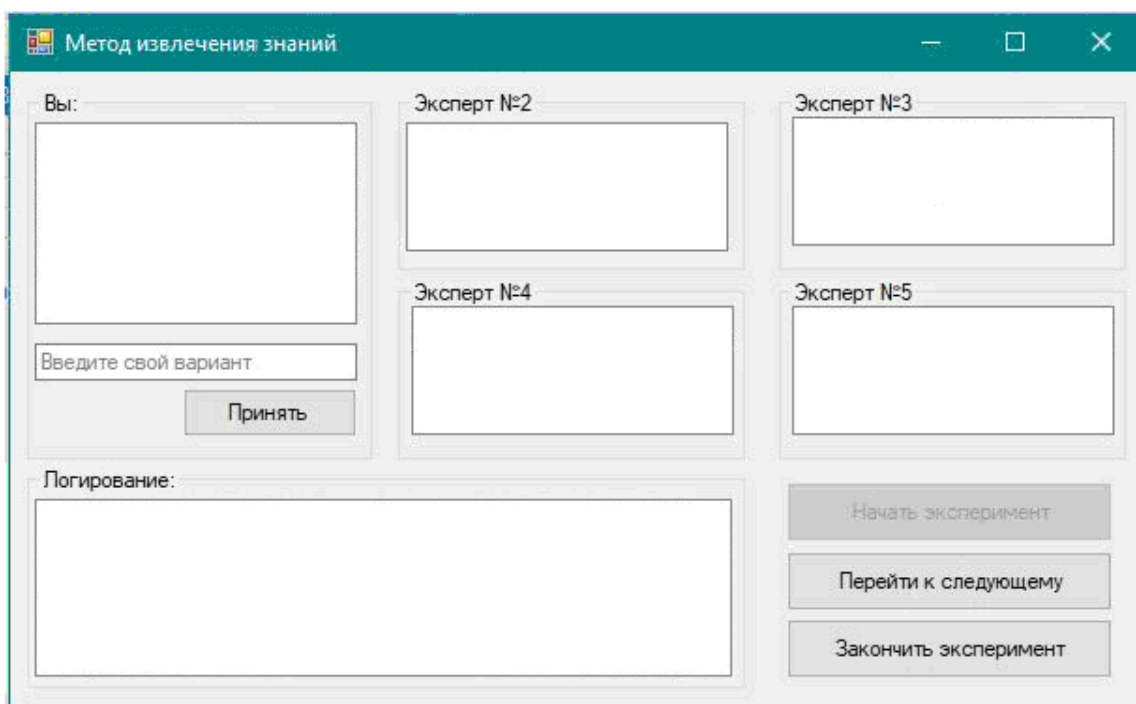


Рисунок 3.8 – Рабочее окно при выборе активного коммуникативного метода

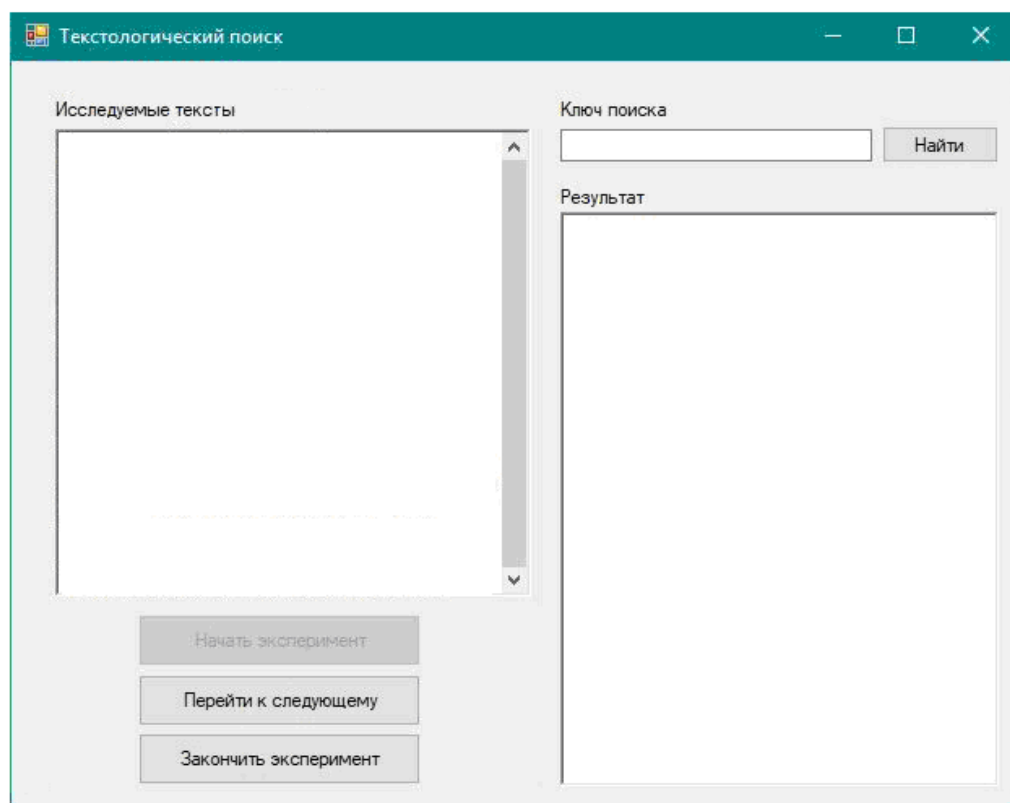


Рисунок 3.9 – Рабочее окно при выборе текстологического метода

- для группы активных коммуникативных методов – возможность для когнитолога вмешаться в процесс извлечения знаний (кнопка «Принять» позволяет когнитологу перевести процесс извлечения знаний в необходимое русло в том случае, если эксперты отходят от выбранной тематики);

- для группы текстологических методов – возможность поиска по ключу (ключевое слова / слова, словосочетания, фрагмент текста и т.д.).

Также на рисунках 3.7 и 3.8 можно увидеть окно «Логирование», в котором отражается весь процесс извлечения знаний выбранным методом с указанием дат и времени работы, а также всех высказываний с указанием авторства, и окна пользователей информационной системы (когнитолога и экспертов), которые в зависимости от количества выбранных экспертов и необходимости участия когнитолога могут быть или активными или отсутствовать.

Для текстологических методов в рабочей зоне окна размещены два поля: для загрузки текстов и для вывода полученных результатов.

3.2 Контрольный пример реализации проекта

В качестве контрольного примера была выбрана реализация процедуры извлечения экспертных знаний комбинированным способом с использованием одной из наиболее часто встречающейся последовательности методов: наблюдение – мозговой штурм – анализ литературы. Выбранная задача – подходы к реализации интеллектуальных систем.

Для начала работы загружается главная страница информационной системы (рис. 3.1). Затем, при нажатии кнопки «Начать эксперимент», подгружается модуль выбора методов извлечения знаний и предлагается

выбрать первый метод, который будет использоваться в процедуре (рис. 3.2, 3.3).

После выбранного первого метода (наблюдение) предлагается выбрать второй метод из выпадающего списка, отсортированного согласно выбранному первому методу (рис. 3.10).

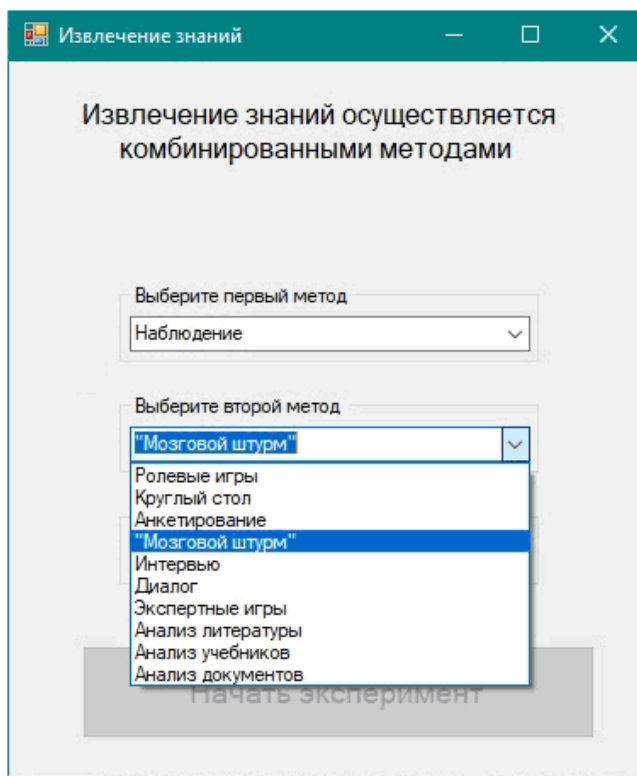


Рисунок 3.10 – Выбран первый метод – наблюдение

На рисунках 3.11 и 3.12 можно видеть выбранные второй и третий метод извлечения знаний соответственно.

Далее при нажатии кнопки «Начать эксперимент» переходим ко второму блоку – непосредственно к процедуре извлечения знаний комбинированным способом с последовательным применением трех выбранных методов.

Вся предварительная работа перед каждым методом, заключающаяся в подготовке материала и ознакомлении с ним в случае необходимости экспертов, в данной ВКР не рассматривается.

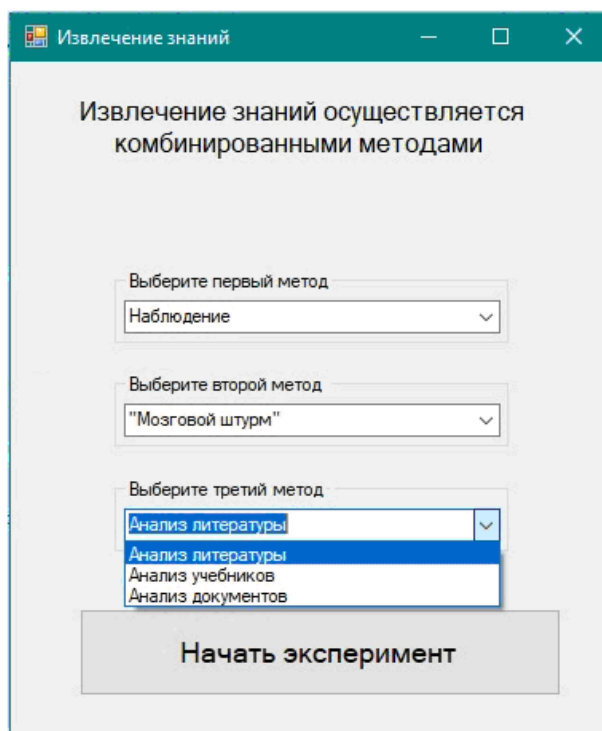


Рисунок 3.11 – Выбран второй метод – мозговой шторм

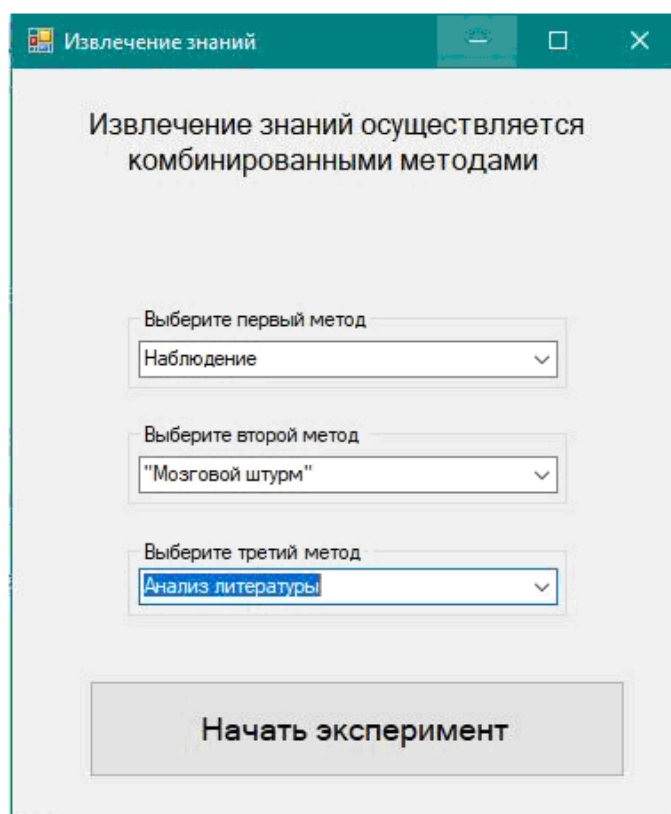


Рисунок 3.12 – Выбран третий метод – анализ литературы

На рисунке 3.13 представлен протокол процесса извлечения знаний методом «Наблюдение». Так как предлагаемая разработка является демонстрационным прототипом, то загрузка в окна экспертов осуществлялась случайно из заранее отобранных высказываний по выбранной задаче. Процесс извлечения знаний прекращается либо при нажатии кнопки «Перейти к следующему» либо по истечении определенного времени, когда эксперты были не активны. В примере переходим нажатием кнопки.

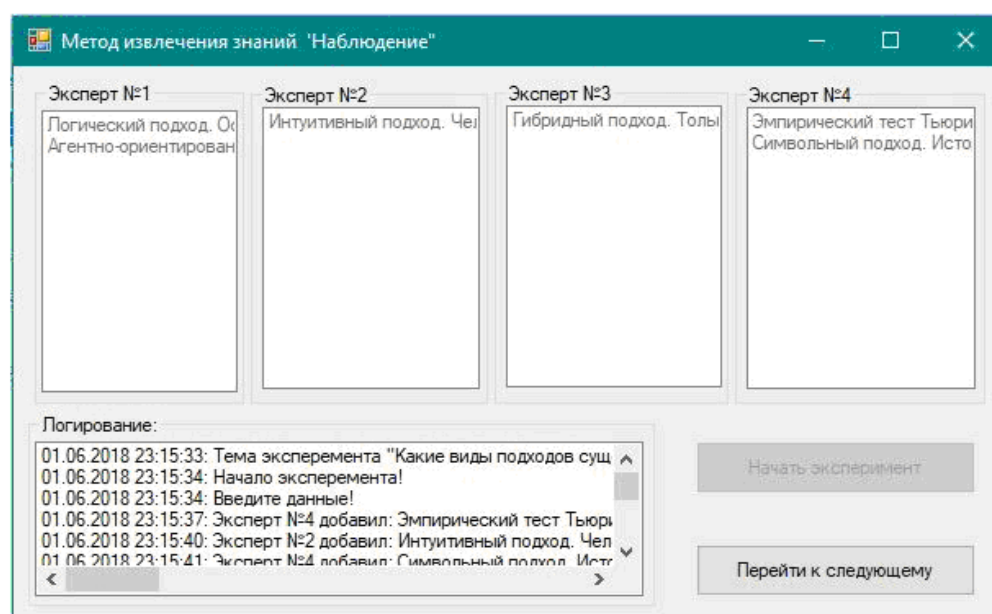


Рисунок 3.13 – Процесс извлечения знаний методом «Наблюдение»

На рисунке 3.14 представлен процесс извлечения знаний методом «Мозговой штурм». В действительности эксперты предлагают свои оригинальные идеи, но в данном примере используется генерация ответов из заранее подобранных. Процесс извлечения знаний прекращается либо при нажатии кнопки «Перейти к следующему» либо по истечении определенного времени, когда эксперты были не активны. В примере переходим нажатием кнопки.

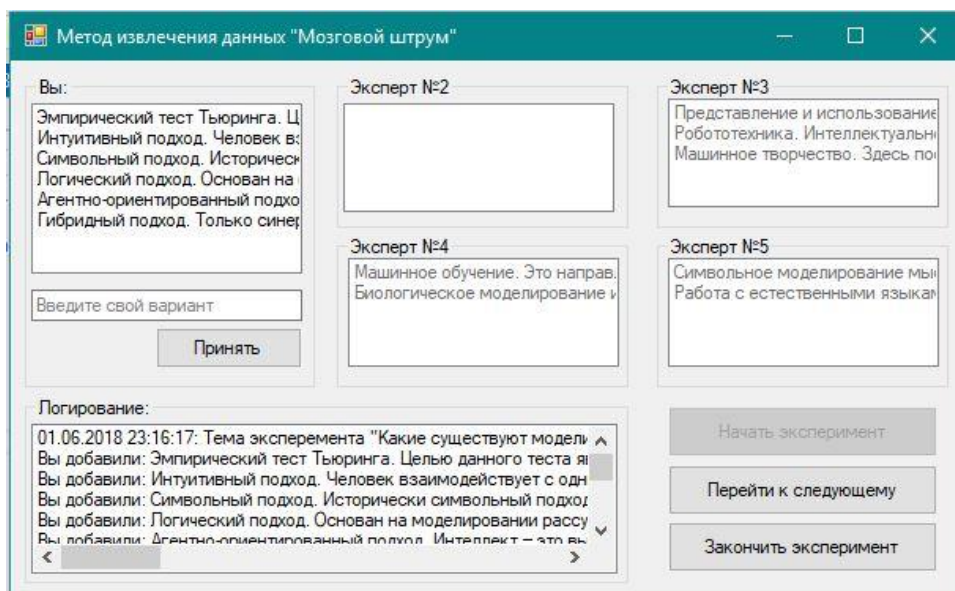


Рисунок 3.14 – Процесс извлечения знаний методом «Мозговой штурм»

На рисунке 3.15 представлен этап загрузки текста / текстов, отобранных заранее по предлагаемой тематике. В действительности можно не ограничиваться подобранными текстами, а осуществлять поиск в любых доступных базах. В контрольном примере была осуществлена загрузка двух статей и одного учебного пособия по выбранной тематике.

На рисунках 3.16 и 3.17 представлены результаты поиска знаний в загруженных текстах по ключевым словам. Ключевые слова можно или внести вручную или загрузить из базы с учетом ранее проведенных процедур извлечения знаний. В примере происходила загрузка выбором из предлагаемого списка.

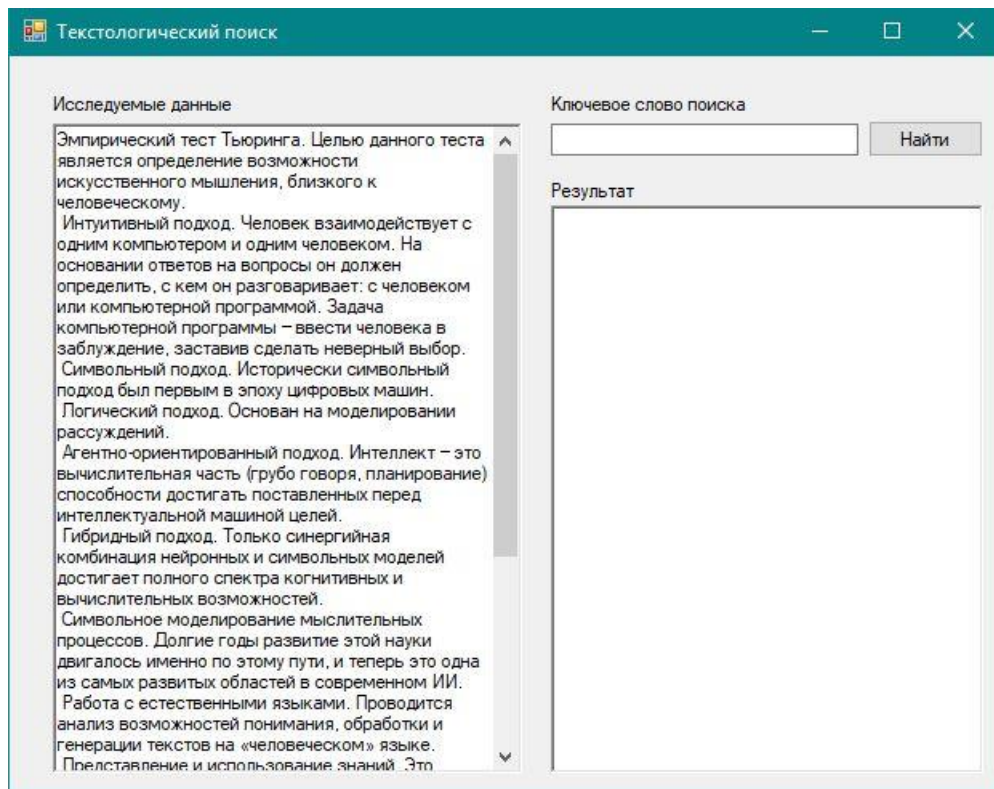


Рисунок 3.15 – Процесс извлечения знаний методом «Анализ литературы».
Загрузка текста

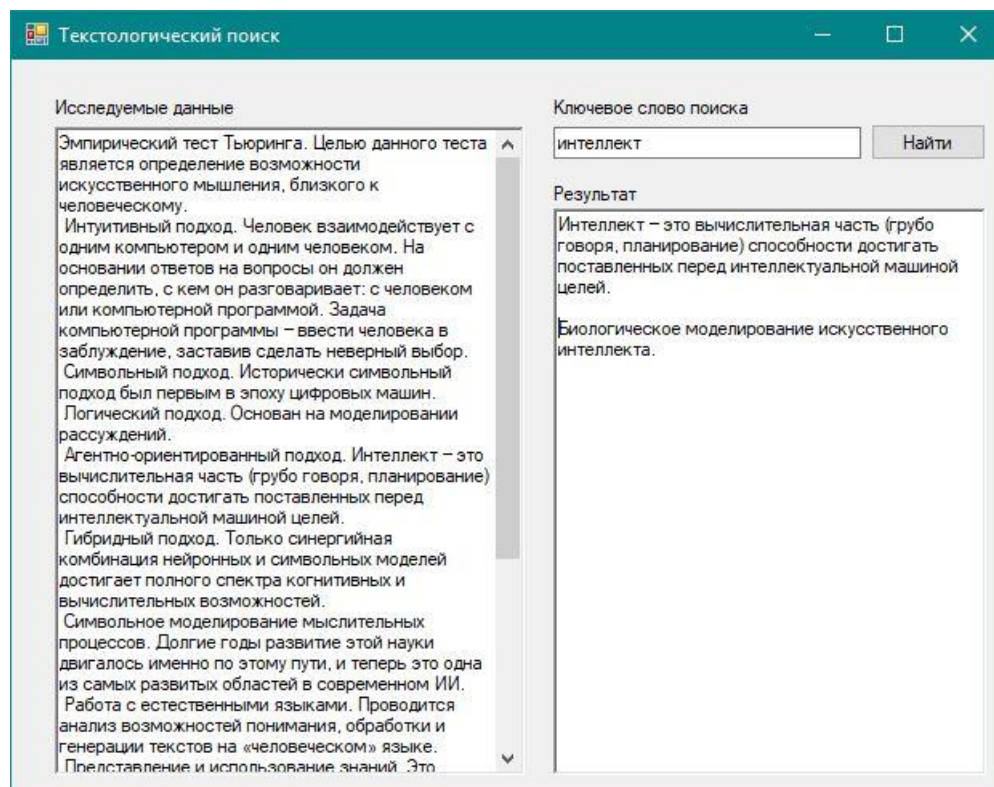


Рисунок 3.16 – Процесс извлечения знаний методом «Анализ литературы».
Поиск по первому ключевому слову

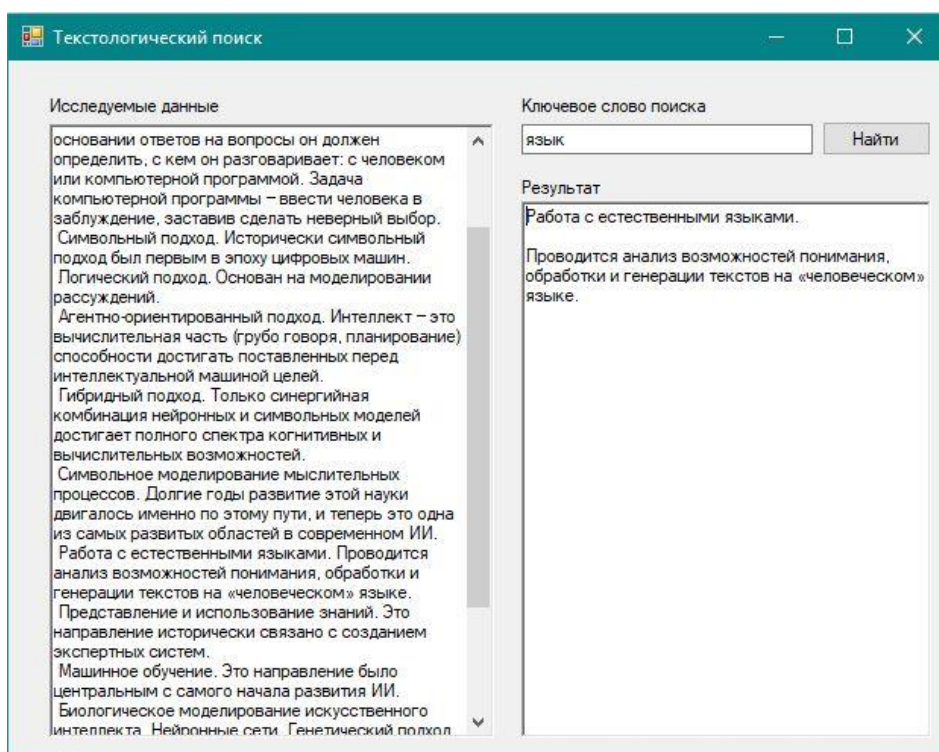


Рисунок 3.17 – Процесс извлечения знаний методом «Анализ литературы».
Поиск по второму ключевому слову

3.3 Анализ социально-экономической эффективности проекта

3.3.1 Определение эффективности предлагаемого подхода

Проектирование и разработка информационной системы поддержки извлечения знаний комбинированными методами позволит снизить трудоемкость работ, повысить производительность труда как когнитолога, так и экспертов.

Цель разработки – сокращение времени работы экспертной группы за счет использования комбинированного метода извлечения знаний.

Для оценки качества работы разработанной системы было проведено тестирование. В процессе тестирования было определено насколько может

быть сокращено время для решения одной задачи, при этом рассматривались три последовательно применяемых метода. Все данные приведены в таблице 3.1. Так как был разработан демонстрационный прототип, то в таблице не рассматриваются вопросы выбора когнитолога, предметной и проблемной области решаемой задачи, формирования базы экспертов, подбор текстовых источников.

Таблица 3.1 – Результаты тестирования

| Этап процедуры | Исполнитель | Примечание | $t_{исп}$, час | $t_{нов}$, час | $t_{нов}-t_{исп}$, час | $t_{нов}/t_{исп}$, час |
|---|--------------------|--|-----------------|-----------------|-------------------------|-------------------------|
| 1) Выбор метода извлечения знаний (наблюдение) | Когнитолог | | 0,2 | 0,1 | -0,1 | 0,5 |
| 2) Выбор эксперта \ экспертов | Когнитолог | | 1 | 0,25 | -0,75 | 0,75 |
| 3) Проведение процедуры извлечения знаний методом «Наблюдение» | Когнитолог | Дополнительное время работы без эксперта | 0,5 | 0 | -0,5 | 0 |
| | Эксперт / эксперты | Старое решение – 1 Новое решение – до 4 | 1 | 1 | 0 | 1 |
| 4) Анализ полученных результатов | Когнитолог | | 3 | 0,5 | -2,5 | 0,17 |
| 5) Принятие решения о выборе следующего метода | Когнитолог | В новом решении – вместе с 1 этапом | 0,25 | 0,05 | -0,2 | 0,2 |
| 6) Выбор метода извлечения знаний (мозговой штурм) | Когнитолог | Одновременно с этапом 1 в новом решении | 0,5 | 0,05 | -0,45 | 0,1 |
| 7) Выбор экспертов | Когнитолог | | 3 | 0,25 | -2,75 | 0,08 |
| 8) Проведение процедуры извлечения знаний методом «Мозговой штурм» | Когнитолог | Дополнительное время работы без эксперта | 4 | 0 | -4 | 0 |
| | Эксперт / эксперты | Старое решение – 1 Новое решение – до 4 | 0,75 | 0,75 | 0 | 1 |
| 9) Анализ полученных результатов | Когнитолог | | 3 | 0,5 | -2,5 | 0,17 |
| 10) Принятие решения о выборе следующего метода | Когнитолог | В новом решении – вместе с 1 этапом | 0,25 | 0,05 | -0,2 | 0,2 |
| 11) Выбор метода извлечения знаний (текстологический метод) | Когнитолог | Одновременно с этапом 1 в новом решении | 0,5 | 0,05 | -0,45 | 0,1 |
| 12) Проведение процедуры извлечения знаний текстологическим методом | Когнитолог | Рассматривались две статьи и одно учебное пособие (пример) | 12 | 0,5 | -11,5 | 0,42 |
| 13) Анализ полученных результатов | Когнитолог | | 3 | 0,5 | -2,5 | 0,17 |
| Итого | | | 32,95 | 4,55 | -28,4 | 0,14 |

Как видно из таблицы для конкретной задачи время выполнения сократилось с 32.95 до 4.55 часов. Графическое представление результатов представлено на рисунке 3.18.

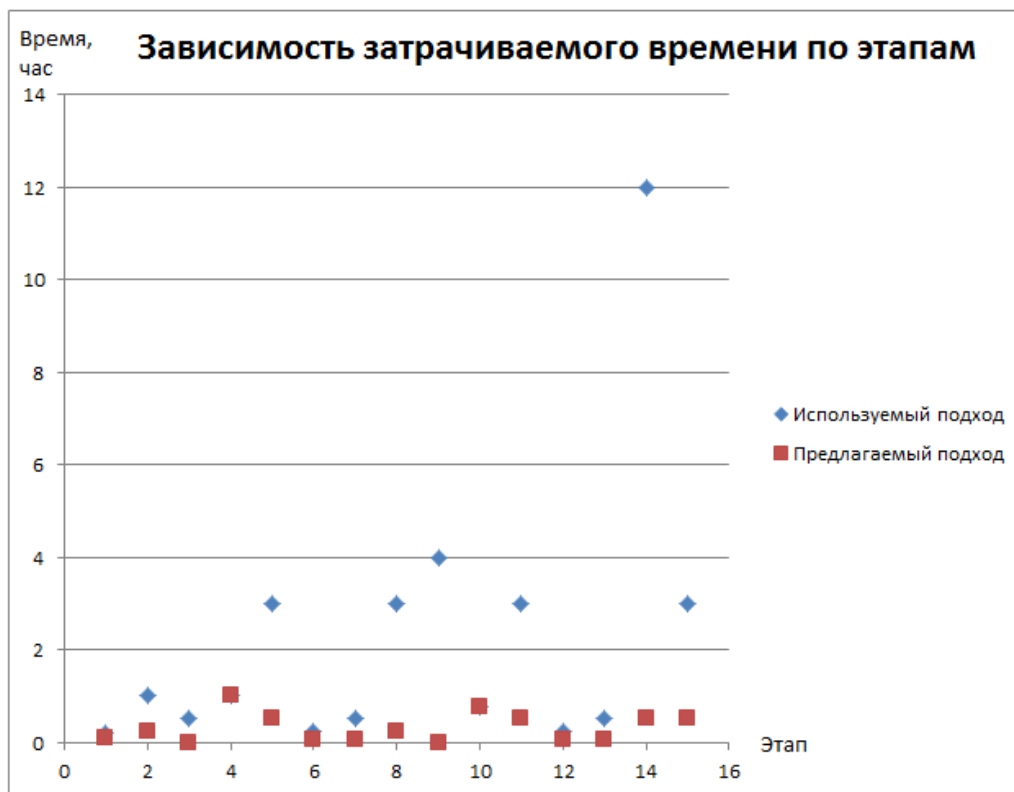


Рисунок 3.18 - Зависимость времени, затрачиваемого по этапам

3.3.2 Расчет затрат на разработку информационной системы

К затратам на разработку и внедрение относятся:

- затраты разработчика на научно исследовательскую работу (НИР), включая затраты на теоретические исследования, подбор и изучение литературы, согласование и утверждение технического задания и т.д. (КНИР);
- затраты на анализ, проектирование, совершенствование, доработку и адаптацию информационной системы;

- затраты организации на внедрение, закупку и монтаж оборудования, обучение персонала и т.д. (КНОВ);
- общие капитальные вложения, включая затраты на НИР и новое оборудование.

В смету затрат на НИР включаются:

- материальные затраты;
- основная и дополнительная зарплата разработчиков;
- отчисления на социальные нужды;
- амортизационные отчисления;
- затраты на эксплуатацию оборудования;
- затраты на научно-техническую информацию;
- затраты на программное обеспечение при использовании ЭВМ;
- накладные расходы.

Оценка трудоемкости выполнения ВКР приведена в таблице 3.2.

Таблица 3.2 – Оценка трудоемкости выполнения ВКР

| Стадии НИР | Основные виды работ | Трудоемкость | |
|-------------------------------|--|--------------|-----|
| | | дни | % |
| Подготовительная | Подбор и изучение литературы. Согласование и утверждения технического задания и календарного плана работ | 3 | 12 |
| Теоретическая разработка темы | Теоретическая проработка вопроса. Постановка задачи | 4 | 16 |
| Практическая реализация | Проектирование и разработка демонстрационного прототипа | 10 | 40 |
| Обобщения и выводы | Обобщения и выводы по проделанной работе | 4 | 16 |
| Техническая отчетность | Подготовка инструкций и отчетов о выполненной работе | 3 | 12 |
| Заключительная стадия | Защита отчета, утверждение результатов | 1 | 4 |
| Итого: | | 25 | 100 |

К материальным затратам относится стоимость сырья, материалов, канцелярских и расходных товаров в действующих ценах, использованных при проведении исследований (таблица 3.3).

Таблица 3.3 – Смета затрат на приобретение покупных комплектующих изделий разработчиком

| Наименование покупных изделий | Марка, тип | Кол-во, шт. | Цена за ед., руб. | Стоимость, руб. |
|--------------------------------|------------------------|-------------|-------------------|-----------------|
| Бумага (упаковка) | Ballet 80 г/м2 А4 500л | 1 | 220 | 220 |
| Расходные материалы на принтер | Canon LBP6020 | 1 | 235 | 235 |
| Ручка шариковая | BIC BPS-GP-F | 3 | 15 | 45 |
| Итого: | | | | 500 |

Основная заработная плата $Z_{осн}$ включает оплату труда разработчика, руководителя и консультантов за период НИР.

$$Z_{осн} = \sum_{i=1}^n T_{об} \cdot Z_{ср.дн.} \cdot i, \quad (3.1)$$

где $T_{об}$ – общая трудоемкость проекта, дни;

$Z_{ср.дн.}$ – среднедневная заработная плата одного работника i -ой категории, руб.

Основная заработная плата разработчика:

$$Z_{осн1} = 300 \text{ руб./день} \cdot 25 \text{ дней} = 7500 \text{ руб.}$$

Основная заработная плата консультанта:

$$Z_{осн2} = 250 \text{ руб./ч.} \cdot 20 \text{ ч.} = 5000 \text{ руб.}$$

Общий фонд заработной платы складывается из основной и дополнительной заработной платы разработчика, и консультанта за период НИР:

$$Z_{сум} = \sum_{i=1}^n (Z_{осн.i} + Z_{доп.i}) \quad (3.2)$$

Общий фонд заработной платы составляет:

$$Z_{сум} = 7500 + 5000 = 12500 \text{ руб.}$$

Отчисления на социальные отчисления принимаются по действующему законодательству на момент написания ВКР 30,2 % к общему фонду заработной платы: 3775 руб.

Амортизационные отчисления использованных в период выполнения НИР оборудования, инструментов, ЭВМ:

$$A_m = \frac{O_{\phi} \cdot H_a \cdot T_m}{365 \cdot 100}, \quad (3.3)$$

где O_{ϕ} – стоимость машин, оборудования, ЭВМ;

H_a – норма амортизации, %;

T_m – время эксплуатации оборудования за период НИР, дни.

Для определения общей суммы амортизационных отчислений по всему используемому и оборудованию составлена смета (таблица 3.4).

Таблица 3.4 – Смета амортизационных отчислений за период НИР

| Вид оборудования | Стоимость, руб. | Срок службы, лет | Годовая норма амортизации, % | Сумма амортизации за период НИР, руб. |
|-----------------------|-----------------|------------------|------------------------------|---------------------------------------|
| Ноутбук ASUS | 30000 | 5 | 20 | 1972,6 |
| Принтер Canon LBP6020 | 4000 | 5 | 20 | 266,67 |
| Сетевой фильтр SVEN | 700 | 5 | 20 | 46,03 |
| Мышь M-U94 | 300 | 5 | 20 | 19,73 |
| Итого: | 35000 | | | 2305,03 |

Затраты на эксплуатацию оборудования (Ноутбук и принтер) включают стоимость электроэнергии (таблица 3.5):

$$Z_{\text{эл}} = C_{\text{эл}} \cdot M_m \cdot T_m \cdot T_{\text{сут}}, \quad (3.4)$$

где $C_{\text{эл}}$ – стоимость 1 кВт/ч электроэнергии, руб.;

M_m – мощность оборудования, кВт/ч;

T_m - время эксплуатации оборудования за период НИР;

$T_{\text{сут}}$ – время работы оборудования в сутки.

Затраты на эксплуатацию оборудования составляют:

$$Z_{\text{эл}} = (3,62 \cdot 0,5 \cdot 25 \cdot 6) + (3,62 \cdot 0,1 \cdot 5 \cdot 1) = 273,31 \text{ руб.}$$

Затраты на научно-техническую информацию учитывают стоимость купленных книг, справочников, оплату ксерокопирования и получения доступа в Интернет (таблица 3.6).

Таблица 3.5 – Параметры эксплуатации оборудования за период НИР

| Параметр | Значение |
|---|-----------|
| Стоимость 1 кВт/ч электроэнергии | 3,62 руб. |
| Мощность ноутбука | 0,5 кВт/ч |
| Мощность принтера | 0,1 кВт/ч |
| Время эксплуатации ноутбука за период НИР | 25 дней |
| Время эксплуатации принтера за период НИР | 5 дней |
| Время работы ноутбука в сутки | 6 часов |
| Время работы принтера в сутки | 1 час |

Таблица 3.6 – Смета затрат на научно-техническую информацию

| Статья затрат | Марка, тип | Количество | Цена за единицу, руб. | Стоимость, руб. |
|------------------|------------|------------|-----------------------|-----------------|
| Ксерокопирование | лист А4 | 50 | 2 | 100 |
| Итого: | | | | 100 |

Затраты на программное обеспечение при использовании ЭВМ включают стоимость программных продуктов на период выполнения НИР. Так как разработка велась с помощью ПО НИУ БелГУ, то к затратам данную статью относить не будем.

Накладные расходы включают затраты на освещение, отопление, арендную плату за помещение (таблица 3.7).

Таблица 3.7 – Смета накладных расходов

| Статья затрат | Марка, тип | Количество | Цена за единицу, руб. | Стоимость, руб. |
|--------------------|------------|------------|-----------------------|-----------------|
| Плата за освещение | кВт/ч | 40 | 3,62 | 144.8 |
| Итого: | | | | 144.8 |

По данным предыдущих расчетов составлена смета затрат на выполнение НИР (таблица 3.8).

Таблица 3.8 – Смета затрат на выполнение НИР

| Элементы затрат | Сумма, руб. |
|--|-------------|
| Материальные затраты | 500 |
| Основная зарплата разработчиков | 12500 |
| Отчисления на социальные нужды | 3775 |
| Амортизационные отчисления | 2305,03 |
| Затраты на эксплуатацию оборудования | 273,31 |
| Затраты на научно-техническую информацию | 100 |
| Накладные расходы | 144,8 |
| Итого: $K_{\text{НИР}}$ | 19598,14 |

В третьем разделе приведено описание разработки демонстрационного прототипа информационной системы извлечения знаний, рассмотрен контрольный пример, приведено социально-экономическое обоснование предлагаемого решения.

ЗАКЛЮЧЕНИЕ

Методы извлечения знаний остаются одним из наиболее актуальных вопросов, которые решаются в рамках развития направления искусственного интеллекта. Построение систем, основанных на человеческих знаниях, будет актуально еще несколько десятков лет. В связи с этим вопросы перенесения процессов извлечения экспертных знаний в компьютерную среду являются актуальными. Используемые сегодня методы извлечения знаний требуют переосмысления в вопросе последовательного применения нескольких методов к решению задачи при условии, что после каждого применимого метода предметная область остается в определенных когнитологом границах, а применяемые решения базируются на ранее полученных.

В ходе выполнения ВКР был разработан демонстрационный прототип информационной системы поддержки извлечения знаний и на контрольном примере определено на сколько сократиться время работы экспертной группы.

Были решены следующие задачи:

- проанализированы существующие методы извлечения знаний;
- проанализированы возможные подходы и способы извлечения знаний;
- построены модели традиционного и предлагаемого процессов извлечения знаний и проанализировать их;
- проведена практическая апробация и проанализирована полученные результаты.

В первом разделе рассмотрена предметная область: изучены вопросы приобретения и извлечения знаний и основные вопросы их развития, выделены основные характеристики процессов извлечения экспертных знаний, построена диаграмма процесса извлечения знаний (как есть), проведен анализ систем поддержки извлечения и приобретения знаний

Во втором разделе рассмотрен процесс проектирования информационной системы извлечения знаний комбинированным методом:

построена диаграмма процесса извлечения экспертных знаний (как должно быть), выявлены наиболее вероятные комбинации последовательно применимых методов извлечения знаний, описан алгоритм извлечения знаний комбинированным методом, построена модель базы данных для информационной системы извлечения знаний.

В третьем разделе приведено описание разработки демонстрационного прототипа информационной системы извлечения знаний, рассмотрен контрольный пример, приведено социально-экономическое обоснование предлагаемого решения.

Вся предварительная работа перед каждым методом, заключающаяся в подготовке материала и ознакомлении с ним в случае необходимости экспертов, в данной ВКР не рассматривается. В дальнейшем планируется доработка демонстрационного прототипа до действующего с разработкой модулей:

- предобработки материала перед каждым используемым методом,
- по отбору и работе с экспертами,
- по формированию предметной области и т.д.

Таким образом, все поставленные задачи решены, цель ВКР – достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 2.105-95 Общие требования к текстовым документам [Текст]. – Введ. 1996–07–01. – М.: Изд-во стандартов, 1995. – 30 с. – (Единая система конструкторской документации)
2. ГОСТ 7.1-2003 Библиографическая запись. Библиографическое описание: общие требования и правила составления [Текст]. – Введ. 2004–07–01. – М.: Изд-во стандартов, 2004. – 105 с. – (Система стандартов по информации, библиотечному и издательскому делу).
3. ГОСТ 7.32-2001 Отчет о научно-исследовательской работе: структура и правила оформления. [Текст]. – Введ. 2002–07–01. – М.: Изд-во стандартов, 2001. – 20 с. – (Система стандартов по информации, библиотечному и издательскому делу)
4. ГОСТ 7.80-2000. Библиографическая запись. Заголовок: Общие требования и правила составления [Текст]. – Введ. 2000–07–01. – М.: Изд-во стандартов, 2000. – 11 с. – (Система стандартов по информации, библиотечному и издательскому делу).
5. ГОСТ 7.0.5-2008 Библиографическая ссылка: общие требования и правила составления [Текст]. – Введ. 2008–05–28. – М.: Изд-во стандартов, 2009. – 23 с. – (Система стандартов по информации, библиотечному и издательскому делу).
6. Александров, Д.В. Инструментальные средства информационного менеджмента. CASE-технологии и распределенные информационные системы: Учебное пособие [Текст] / Д.В. Александров. – М.: ФиС, 2011. – 224 с.
7. Балдин К.В. Информационные системы в экономике [Текст]/ К.В.Балдин, В.Б.Уткин. – М.: Дашков и К, 2010. – 395 с.
8. Гаврилова, Т.А. Базы знаний интеллектуальных систем [Электронный ресурс] / Т.А. Гаврилова, В.Ф.Хорошевский. - Ссылка: <http://webrabota.com/download/samouchiteli/gavrilova-ta-khoroshevskij->

vf-bazy-znaniy-intellektualnyh-sistem-2000 (дата обращения 12.12.2017).

9. Гаврилова, Т.А. От инженерии знаний к онтологическому инжинирингу [Электронный ресурс] / Т.А. Гаврилова. - Ссылка: <http://posp.raai.org/data/posp2005/gavrilova/gavrilova.html> (дата обращения 12.04.2018).

10. Гаврилова, Т.А. Инженерия знаний. Модели и методы [Электронный ресурс] / Т.А. Гаврилова, Д.И. Кудрявцев. - Ссылка: <http://mirknig.su/knigi/programming/73955-inzheneriya-znaniy-modeli-i-metody.html> (дата обращения 12.04.2018).

11. Глухих, И.Н. Интеллектуальные информационные системы : учеб. пособие для студентов учреждений высш. проф. образования [Текст] / И. Н. Глухих ; М- во образования и науки Рос. Федерации, Тюм. гос. ун-т. - М. : Академия, 2012. - 110 с.

12. Горбенко А.О. Информационные системы в экономике Учебное пособие [Текст] / А.О.Горбенко. – СПб.: Бином. Лаборатория знаний, 2010. – 292 с.

13. Гусятников, В.Н. Стандартизация и разработка программных систем [Текст]/ В.Н. Гусятников, А.И. Безруков. – М.: Финансы и статистика, 2012. – 288 с.

14. Дубейковский, В. И. Практика функционального моделирования с AllFusion Process Modeler 4.1. (ВРwin) Где? Зачем? Как? [Текст]/ В.И. Дубейковский. – М.: ДИАЛОГ-МИФИ, 2014. – 464 с.

15. Дуда, Р. Распознавание образов и анализ сцен [Текст] / Р. Дуда, П. Харт. - М.: Мир, 2013.

16. Ефимов, Е. Н. Информационные системы и технологии в экономике : учеб. пособие для студентов, обучающихся по специальностям "Мировая экономика", "Бухгалт. учет, анализ и аудит" и "Финансы и кредит" [Текст] / Е. Н. Ефимов, Е. В. Ефимова, Г. М. Лапицкая ; под ред. Г. М. Лапицкой. - Ростов н/Д : МарТ: Феникс, 2012. - 296 с.

17. Комаров, С.И. Архитектура системы SIMER [Текст] / С.И. Комаров,

Д.Е. Малаховский, Г.И. Назаренко и др. // Тез. докл. Всесоюз. конф. по искусственному интеллекту. Т. 3. - М.: НС «Искусственный интеллект» АН СССР, 1988. - С. 501-502.

18. Кук, П.М. Формальная методология приобретения и представления экспертных знаний [Текст] / П.М. Кук, Д.Э. Макдональд // ТИИЭР. 1986.Т. 74. - № 10. - С. 145-155.

19. Ларичев, О.И. Системы поддержки принятия решений для слабоструктурированных проблем: требования и ограничения [Текст] / О.И. Ларичев, А.Б. Петровский //Человеко-машинные процедуры принятия решений: Сб. 1тр. - М.: ВНИИСИ, 1988. - С. 4-13.

20. Люггер, Дж.Ф. Искусственный интеллект. Стратегии и методы решения сложных проблем / Дж.Ф. Люггер. - Москва, Санкт-Петербург, Киев: Издательство «Вильямс» – 2016.

21. Маклаков, С.В. BPwin и ERwin. CASE – средства разработки информационных систем [Текст] / С.В. Маклаков. – М.: Диалог-МИФИ, 2014. – 256 с.

22. Маклаков, С.В. Моделирование бизнес-процессов с AllFusion Process Modeler [Текст] / С.В. Маклаков. – М.: Диалог-МИФИ, 2014. – 224 с.

23. Моргоев, В.К. Интерактивный метод извлечения экспертных знаний на основе имитации консультаций [Текст] / В.К. Моргоев // Тез. докл. Всесоюз. конф. по искусственному интеллекту. Т. 3. - М.: НС «Искусственный интеллект» АН СССР, 1988. - С. 318-351.

24. Попов, Э.В. Экспертные системы. Решение неформализованных задач в диалоге с ЭВМ [Текст] / Э.В. Попов. - М.: Наука, 2015.

25. Терехина, А.Ю. Анализ данных методами многомерного шкалирования [Текст] / А.Ю. Терехина. - М.: Наука, 2016.

26. Форсайт, Р. Экспертные системы. Принципы работы и примеры [Текст] / Р. Форсайт. - М.: Радио и связь, 2014.

27. Уотерман, П. Руководство по экспертным системам [Текст] / П. Уотерман. - М.: Мир, 2016.

28. Элти, Дж. Экспертные системы: концепции и примеры [Текст] / Дж. Элти, М.Кумбс. - М.: Финансы и статистика, 2013.
29. Ясницкий, Л. Н. Введение в искусственный интеллект : учеб. пособие для студентов вузов, обучающихся по мат. направлениям и специальностям [Текст] / Л. Н. Ясницкий. - 3-е изд., стер. - М. : Академия, 2013. - 175 с.
30. Bennet, J.S. ROGET: A knowledge-based system for acquiring the conceptual structure of a diagnostic expert system [Текст] / J.S. Bennet //J. Automated Reasoning. 1985. - V. 1. - P. 19-74.
31. Boose J.H. Expertise transfer for expert system design [Текст] / J.H. Boose. - N. Y.: Elsevier, 2013.
32. Boose, J.H. Expertise transfer and complex problems: using AQUINAS as a knowledge-acquisition workbench for knowledge-based systems [Текст] / J.H. Boose, J.M. Bradsaw // International J. of Man-Machine Studies. 1987. V. 26. № 1. P. 3-28.
33. Davis, R. Knowledge-based systems in artificial intelligence [Текст] / R. Davis, D.B. Lenat. - N. Y.: McGraw-Hill Inc., 2012.
34. Doukidis, G.I. Developing Expert Systems [Текст] / G.I. Doukidis, E.H. Whitley. - London: Chartwell-Bratt, 1988.
35. Ericsson, K.A. Verbal reports as data [Текст] / K.A. Ericsson, H.A. Simon //Psychological Review. 1980. V. 82. JNs 2. P. 215-251.
36. Hawkins, D. An analysis of expert thinking / D. Hawkins //International J. Man-Machine Studies. 1983. V. 18. P. 1-47.
37. Hart, A. Knowledge elicitation: issues and methods / A. Hart //Computer-Aided Design. 1985. V. 18. № 9. P. 455-462.
38. Hart, A. Knowledge acquisition for expert systems / A. Hart. - London: Kogan Page, 2013.
39. Hickman, F. Knowledge acquisition: the key to success for commercial expert systems / F. Hickman // KBS86. Knowledge Based Systems. Proc. of the International Conference, London, July. 1986. P. 205-214.
40. Fisher, D.H. Knowledge Acquisition Via Incremental Conceptual

Clustering [Электронный ресурс] / D.H. Fisher. // Machine Learning. Статья в Интернет: <http://www.springerlink.com/content/qj16212n7537n6p3/fulltext.pdf> (дата обращения 05.04.2018)

41. Johnson, L. Expert systems technology / L. Johnson, E.T. Keravnou. - Abacus Press, 2012.

42. Kahn, G. MORE: an intelligent knowledge acquisition tool / G. Kahn, S. Nowlan, J. McDermott //Proc. of Ninth Intern. Conf. on Artif. Intelligence. Los Angeles, California. 1985.

43. McLaren, R. Knowledge acquisition by computer induction / R. McLaren // R&D Management. 1985. V. 15. № 2. P. 159-166.

44. Simon, H.A. The new science of management decision englewood cliffs / H.A. Simon.- New Jersey: Prentice Hall, Inc., 1975.

45. Simon, H.A. Information-processing theory of human problem solving / H.A. Simon // Handbook of learning and cognitive processing. V. 5. Human Information Processing. Ed. Ectes, LEA, 1978. P. 271-295.

46. Waterman, D.A. Protocol analysis as a task for artificial intelligence / D.A. Waterman, A. Newell // Artificial Intelligence. 1971. V. 2. P. 285-318.

47. Wielinga, B.J. Interpretation models for knowledge acquisition / B.J. Wielinga, J.A. Breuker //Proc. Advances in Artificial Intelligence (ECAI, Pisa), North-Holland, Amsterdam, 1984.

ПРИЛОЖЕНИЕ

Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace diloro
{
    public partial class Form1 : Form
    {
        FormMethod1 fm1 = new FormMethod1();
        public Form1()
        {
            InitializeComponent();

            groupBox1.Visible = false;
            groupBox2.Visible = false;
            groupBox3.Visible = false;

            button3.Visible = true;
            button1.Enabled = false;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            fm1.Show();
            Close();
        }

        private void label6_Click(object sender, EventArgs e)
        {
        }

        private void label4_Click(object sender, EventArgs e)
        {
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }

        private void groupBox1_Enter(object sender, EventArgs e)
        {
        }
    }
}
```

```

Font myFont = new Font("Arial", 10, FontStyle.Regular);
private void button3_Click(object sender, EventArgs e)
{
    button3.Visible = false;
    groupBox1.Visible = true;
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    groupBox2.Visible = true;
}

private void comboBox2_SelectedIndexChanged(object sender, EventArgs e)
{
    groupBox3.Visible = true;
}

private void comboBox3_SelectedIndexChanged(object sender, EventArgs e)
{
    button1.Enabled = true;
}
private void comboBox2_DrawItem_1(object sender, DrawItemEventArgs e)
{
    if (e.Index == 0)
    {
        e.Graphics.DrawString(comboBox1.Items[e.Index].ToString(), myFont,
Brushes.LightGray, e.Bounds);
    }
    else
    {
        e.DrawBackground();
        e.Graphics.DrawString(comboBox1.Items[e.Index].ToString(),
myFont, Brushes.Black, e.Bounds);
        e.DrawFocusRectangle();
    }
}
}
}

```

FormMethod1

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace diloro
{
    public partial class FormMethod1 : Form
    {
        //FormMethod2 fm2 = new FormMethod2();

        // private void button1_Click(object sender, EventArgs e)
        // {

```

```

//      fm2.Show();
//      Close();
//      }

List<string> data_set = new List<string>();
List<string> DATA = new List<string>();
List<bool> bot_leave = new List<bool>();
int length_data_set = 0;
Random rand = new Random();
List<string> result_list = new List<string>();

public FormMethod1()
{
    InitializeComponent();
    FillDataSet();
    listBox6.Items.Add(DateTime.Now.ToString() + ": Тема эксперимента \"Какие
виды подходов существуют в разработке искусственного интеллекта?\"!");
}

private bool CheckValueBot(ListBox lb, string value)
{
    for (int i = 0; i < lb.Items.Count; i++)
    {
        if (value.ToLower() == lb.Items[i].ToString().ToLower())
            return true;
    }
    return false;
}

#region Event
//Обработка отправки

//Обработка входа на вход

//Обработка выхода

//Загрузка формы
private void FormMethod1_Load(object sender, EventArgs e)
{
}
#endregion

#region FillData
private void FillDataSet()
{
    StreamReader sr = new StreamReader("Тест.txt",
System.Text.Encoding.GetEncoding(1251));

    while (!sr.EndOfStream)
    {
        data_set.Add(sr.ReadLine());
    }
}
}

```

```

    }
    length_data_set = data_set.Count;
    for (int i = 0; i < 4; i++)
    {
        bot_leave.Add(false);
    }
    sr.Close();
}

private void Logging(string text)
{
    listBox6.Invoke(new Action(() =>
    {
        listBox6.Items.Add(DateTime.Now.ToString() + ": " + text);
    }));
}
#endregion

#region Imitation

private void ImitationTime(ListBox lb, int millisecond, int index)
{
    Stopwatch stopWatch = new Stopwatch();
    stopWatch.Start();
    double temp = 0.5;
    while (stopWatch.ElapsedMilliseconds < millisecond)
    {
        if (stopWatch.ElapsedMilliseconds / (double)1000 > temp)
        {
            if (lb.Items[index].ToString().Length >= 3)
            {
                lb.Invoke(new Action(() =>
                {
                    lb.Items[index] = ".";
                }));
            }
            else
            {
                lb.Invoke(new Action(() =>
                {
                    lb.Items[index] += ".";
                }));
            }
            temp += 0.5;
        }
    }
    stopWatch.Stop();
}

private bool CheckLeaveBot(Random rand, int index_bot)
{
    double coef = 0;
    if (data_set.Count != 0)
        coef = (double)data_set.Count / length_data_set + 0.33;
    if (rand.NextDouble() > coef || data_set.Count == 0)
    {
        bot_leave[index_bot] = true;
        //bot вышел
    }
}

```

```

        if (data_set.Count > 0)
        {
            bool is_all_true = true;
            for (int i = 0; i < 4; i++)
            {
                if (!bot_leave[i])
                    is_all_true = false;
            }
            if (is_all_true)
            {
                bot_leave[index_bot] = false;
            }
        }
        return bot_leave[index_bot];
    }
    return false;
}

private void ImitationBot(int index_bot, ListBox lb, int millisecond, Random
rand, int length_data, bool leave = false)
{
    if (bot_leave[index_bot])
        return;

    lb.Invoke(new Action(() =>
    {
        lb.Items.Add("");
    }));
    int index = lb.Items.Count - 1;
    //Время ...
    ImitationTime(lb, millisecond, index);

    if (CheckLeaveBot(rand, index_bot))
    {
        lb.Invoke(new Action(() =>
        {
            lb.Items.RemoveAt(index);
            lb.Enabled = false;
        }));
        Logging("Эксперт №" + (index_bot + 1).ToString() + " покинул
эксперимент!");
        return;
    }

    if (0 < data_set.Count)
    {
        bool chek = false;
        if (!chek)
        {
            string add_item = data_set[0];
            lb.Invoke(new Action(() =>
            {
                lb.Items[index] = data_set[0];

                result_list.Add(DateTime.Now.ToString() + ": " + data_set[0]);

            }));
            DATA.Add(data_set[0] + "\n");
            data_set.RemoveAt(0);
        }
    }
}

```

```

        Logging("Эксперт №" + (index_bot + 1).ToString() + " добавил: " +
add_item + "!");
    }
}
if (data_set.Count == 0)
{
    lb.Invoke(new Action(() =>
    {
        lb.Enabled = false;
    }));
}

#region Imitation
private void CheckEnd()
{
    if (!listBox2.Enabled && !listBox3.Enabled && !listBox4.Enabled &&
!listBox4.Enabled)
    {
        button3.Invoke(new Action(() =>
        {
            button3.Enabled = true;
        }));
    }
}

private void Imitation1()
{
    int second = 10000;
    int length_input_data = 0;

    while (data_set.Count > 0 && !bot_leave[0])
    {
        int time_wait = Convert.ToInt32(rand.NextDouble() * second);
        if (!button3.Enabled)
            ImitationBot(0, listBox2, time_wait, rand, length_input_data);
        else
            break;
    }
    CheckEnd();
}

private void Imitation2()
{
    int second = 10000;
    int length_input_data = 0;

    while (data_set.Count > 0 && !bot_leave[1])
    {
        int time_wait = Convert.ToInt32(rand.NextDouble() * second);
        if (!button3.Enabled)
            ImitationBot(1, listBox3, time_wait, rand, length_input_data);
        else
            break;
    }
    CheckEnd();
}

private void Imitation3()
{

```

```

int second = 10000;
int length_input_data = 0;

while (data_set.Count > 0 && !bot_leave[2])
{
    int time_wait = Convert.ToInt32(rand.NextDouble() * second);
    if (!button3.Enabled)
        ImitationBot(2, listBox4, time_wait, rand, length_input_data);
    else
        break;
}
CheckEnd();
}

private void Imitation4()
{
    int second = 5000;
    int length_input_data = 0;

    while (data_set.Count > 0 && !bot_leave[3])
    {
        int time_wait = Convert.ToInt32(rand.NextDouble() * second);
        if (!button3.Enabled)
            ImitationBot(3, listBox5, time_wait, rand, length_input_data);
        else
            break;
    }
    CheckEnd();
}
}
#endregion

private void Imitation()
{
    int second = 500;
    int length_input_data = 0;

    Random rand = new Random();

    int time_wait = Convert.ToInt32(rand.NextDouble() * second);
    ImitationBot(0, listBox2, time_wait, rand, length_input_data);

    time_wait = Convert.ToInt32(rand.NextDouble() * second);
    ImitationBot(1, listBox3, time_wait, rand, length_input_data);

    time_wait = Convert.ToInt32(rand.NextDouble() * second);
    ImitationBot(2, listBox4, time_wait, rand, length_input_data);

    time_wait = Convert.ToInt32(rand.NextDouble() * second);
    ImitationBot(3, listBox5, time_wait, rand, length_input_data);
}

private void EndImitation()
{
    bool is_all_true = false;
    for (int i = 0; i < 4; i++)
    {
        if (!bot_leave[i])
            is_all_true = true;
    }
    while (data_set.Count >= 0 && is_all_true)

```

```

        {
            Imitation();
            is_all_true = false;
            for (int i = 0; i < 4; i++)
            {
                if (!bot_leave[i])
                    is_all_true = true;
            }
        }
        button3.Invoke(new Action(() =>
        {
            button3.Enabled = true;
        }));

        //Вывод
    }
#endregion

private void button3_Click(object sender, EventArgs e)
{
    Close();
}

public List<string> GetData()
{
    return DATA;
}

private void button4_Click(object sender, EventArgs e)
{
    //button1.Enabled = true;
    listBox6.Items.Add(DateTime.Now.ToString() + ": Начало эксперимента!");
    listBox6.Items.Add(DateTime.Now.ToString() + ": Введите данные!");

    Thread th = new Thread(Imitation1);
    th.Start();

    th = new Thread(Imitation2);
    th.Start();

    th = new Thread(Imitation3);
    th.Start();

    th = new Thread(Imitation4);
    th.Start();
    button4.Enabled = false;
}
}
}

```

FormMethod2

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Drawing;
using System.IO;
using System.Linq;

```



```

using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace diloro
{
    public partial class FormMethod2 : Form
    {

        List<string> DATA = new List<string>();
        List<string> data_set = new List<string>();
        List<bool> bot_leave = new List<bool>();
        int length_data_set = 0;
        bool end_exp = false;
        Random rand = new Random();
        List<string> result_list = new List<string>();

        public FormMethod2(List<string> data)
        {
            InitializeComponent();

            listBox6.Items.Add(DateTime.Now.ToString() + ": Тема эксперимента \"Какие существуют модели и методы исследований искусственного интеллекта?\"!");

            foreach (string str in data)
            {
                listBox1.Items.Add(str);
                listBox6.Items.Add("Вы добавили: " + str);
                DATA.Add(str);
            }

            FillDataSet();
            button1.Enabled = false;
        }

        private bool CheckValue(ListBox lb)
        {
            for (int i = 0; i < lb.Items.Count; i++)
            {
                if (textBox1.Text.ToLower() == lb.Items[i].ToString().ToLower())
                {
                    textBox1.Text = "Такое значение уже существует";
                    textBox1.ForeColor = Color.Gray;
                    return true;
                }
            }
            return false;
        }

        private bool CheckValueBot(ListBox lb, string value)
        {
            for (int i = 0; i < lb.Items.Count; i++)
            {
                if (value.ToLower() == lb.Items[i].ToString().ToLower())
                    return true;
            }
            return false;
        }
    }
}

```

```

}

#region Event
//Обработка отправки
private void button1_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "" || textBox1.Text == "Введите свой вариант" ||
        textBox1.Text == "Такое значение уже существует")
        return;

    if (CheckValue(listBox1) || CheckValue(listBox2) || CheckValue(listBox3) ||
        CheckValue(listBox4) || CheckValue(listBox5))
        return;

    for (int i = 0; i < data_set.Count; i++)
    {
        if (textBox1.Text == data_set[i])
        {
            DATA.Add(data_set[i]);
            data_set.RemoveAt(i);
        }
    }

    listBox1.Items.Add(textBox1.Text);

    Logging("Вы добавили " + textBox1.Text + "!");
    DATA.Add(textBox1.Text + "\n");
    result_list.Add(DateTime.Now.ToString() + ": " + textBox1.Text);

    textBox1.Text = "Введите свой вариант";
    textBox1.ForeColor = Color.Gray;
}

//Обработка входа на вход
private void textBox1_Click(object sender, EventArgs e)
{
    if (textBox1.Text == "Введите свой вариант" || textBox1.Text == "Такое
значение уже существует")
        textBox1.Text = "";
    textBox1.ForeColor = Color.Black;
}

//Обработка выхода
private void textBox1_Leave(object sender, EventArgs e)
{
    if (textBox1.Text == "")
    {
        textBox1.Text = "Введите свой вариант";
        textBox1.ForeColor = Color.Gray;
    }
}

//Загрузка формы
private void FormMethod2_Load(object sender, EventArgs e)
{
    textBox1.ForeColor = Color.Gray;
}

```

```

    }
    #endregion

    #region FillData
    private void FillDataSet()
    {
        StreamReader sr = new StreamReader("Tect2.txt",
System.Text.Encoding.GetEncoding(1251));

        while (!sr.EndOfStream)
        {
            data_set.Add(sr.ReadLine());
        }
        length_data_set = data_set.Count;
        for (int i = 0; i < 4; i++)
        {
            bot_leave.Add(false);
        }
        sr.Close();
    }

    private void Logging(string text)
    {
        listBox6.Invoke(new Action(() =>
        {
            listBox6.Items.Add(DateTime.Now.ToString() + ": " + text);
        }));
    }
    #endregion

    #region Imitation

    private void ImitationTime(ListBox lb, int millisecond, int index)
    {
        Stopwatch stopWatch = new Stopwatch();
        stopWatch.Start();
        double temp = 0.5;
        while (stopWatch.ElapsedMilliseconds < millisecond)
        {
            if (stopWatch.ElapsedMilliseconds / (double)1000 > temp)
            {
                if (lb.Items[index].ToString().Length >= 3)
                {
                    lb.Invoke(new Action(() =>
                    {
                        lb.Items[index] = ".";
                    }));
                }
                else
                {
                    lb.Invoke(new Action(() =>
                    {
                        lb.Items[index] += ".";
                    }));
                }
                temp += 0.5;
            }
        }
    }
}

```

```

        stopWatch.Stop();
    }

private bool CheckLeaveBot(Random rand, int index_bot)
{
    double coef = 0;
    if (data_set.Count != 0)
        coef = (double)data_set.Count / length_data_set + 0.33;
    if (listBox1.Items.Count >= 2 && rand.NextDouble() > coef || data_set.Count
== 0)
    {
        bot_leave[index_bot] = true;
        //bot вышел
        if (data_set.Count > 0)
        {
            bool is_all_true = true;
            for (int i = 0; i < 4; i++)
            {
                if (!bot_leave[i])
                    is_all_true = false;
            }
            if (is_all_true)
            {
                bot_leave[index_bot] = false;
            }
        }
        return bot_leave[index_bot];
    }
    return false;
}

private void ImitationBot(int index_bot, ListBox lb, int millisecond, Random
rand, int length_data, bool leave = false)
{
    if (bot_leave[index_bot])
        return;

    lb.Invoke(new Action(() =>
    {
        lb.Items.Add("");
    }));
    int index = lb.Items.Count - 1;
    //Время ...
    ImitationTime(lb, millisecond, index);

    if (CheckLeaveBot(rand, index_bot))
    {
        lb.Invoke(new Action(() =>
        {
            lb.Items.RemoveAt(index);
            lb.Enabled = false;
        }));
        Logging("Эксперт №" + (index_bot + 2).ToString() + " покинул
эксперимент!");
        return;
    }

    if (0 < data_set.Count)
    {
        bool chek = true;

```

```

while (data_set.Count > 0 && chek)
{
    if (CheckValueBot(listBox1, data_set[0]))
    {
        DATA.Add(data_set[0] + "\n");
        data_set.RemoveAt(0);
    }else
    {
        chek = false;
    }
}
if (!chek)
{
    string add_item = data_set[0];
    lb.Invoke(new Action(() =>
    {
        lb.Items[index] = data_set[0];

        result_list.Add(DateTime.Now.ToString() + ": " + data_set[0]);

    }));
    DATA.Add(data_set[0] + "\n");
    data_set.RemoveAt(0);
    Logging("Эксперт №" + (index_bot + 2).ToString() + " добавил " +
add_item + "!");
}
}
if (data_set.Count == 0)
{
    lb.Invoke(new Action(() =>
    {
        lb.Enabled = false;
    }));
}
}

#region Imitation
private void CheckEnd()
{
    if (!listBox2.Enabled && !listBox3.Enabled && !listBox4.Enabled &&
!listBox4.Enabled)
    {
        button3.Invoke(new Action(() =>
        {
            button3.Enabled = true;
        }));
    }
}

private void Imitation1()
{
    int second = 10000;
    int length_input_data = listBox1.Items.Count;

    while (data_set.Count > 0 && !bot_leave[0])
    {
        int time_wait = Convert.ToInt32(rand.NextDouble() * second);
        if (!button3.Enabled)
            ImitationBot(0, listBox2, time_wait, rand, length_input_data);
    }
}

```

```

        CheckEnd();
    }

private void Imitation2()
{
    int second = 10000;
    int length_input_data = listBox1.Items.Count;

    while (data_set.Count > 0 && !bot_leave[1])
    {
        int time_wait = Convert.ToInt32(rand.NextDouble() * second);
        if (!button3.Enabled)
            ImitationBot(1, listBox3, time_wait, rand, length_input_data);
    }
    CheckEnd();
}

private void Imitation3()
{
    int second = 10000;
    int length_input_data = listBox1.Items.Count;

    while (data_set.Count > 0 && !bot_leave[2])
    {
        int time_wait = Convert.ToInt32(rand.NextDouble() * second);
        if (!button3.Enabled)
            ImitationBot(2, listBox4, time_wait, rand, length_input_data);
    }
    CheckEnd();
}

private void Imitation4()
{
    int second = 10000;
    int length_input_data = listBox1.Items.Count;

    while (data_set.Count > 0 && !bot_leave[3])
    {
        int time_wait = Convert.ToInt32(rand.NextDouble() * second);
        if (!button3.Enabled)
            ImitationBot(3, listBox5, time_wait, rand, length_input_data);
    }
    CheckEnd();
}
}
#endregion

private void Imitation()
{
    int second = 1000;
    int length_input_data = listBox1.Items.Count;

    Random rand = new Random();

    int time_wait = Convert.ToInt32(rand.NextDouble() * second);
    ImitationBot(0, listBox2, time_wait, rand, length_input_data);

    time_wait = Convert.ToInt32(rand.NextDouble() * second);
    ImitationBot(1, listBox3, time_wait, rand, length_input_data);

    time_wait = Convert.ToInt32(rand.NextDouble() * second);

```

```

ImitationBot(2, listBox4, time_wait, rand, length_input_data);

time_wait = Convert.ToInt32(rand.NextDouble() * second);
ImitationBot(3, listBox5, time_wait, rand, length_input_data);

if (!end_exp)
{
    button1.Invoke(new Action(() =>
    {
        button1.Enabled = true;
    }));

    button2.Invoke(new Action(() =>
    {
        button2.Enabled = true;
    }));

    Logging("Введите данные!");
}
}

private void EndImitation()
{
    bool is_all_true = false;
    for (int i = 0; i < 4; i++)
    {
        if (!bot_leave[i])
            is_all_true = true;
    }
    while (data_set.Count >= 0 && is_all_true)
    {
        Imitation();
        is_all_true = false;
        for (int i = 0; i < 4; i++)
        {
            if (!bot_leave[i])
                is_all_true = true;
        }
    }
    button3.Invoke(new Action(() =>
    {
        button3.Enabled = true;
    }));

    //Вывод
}
#endregion

private void button2_Click(object sender, EventArgs e)
{
    end_exp = true;
    Logging("Вы покинули эксперимент!");
    button1.Enabled = false;
    button2.Enabled = false;
    //Thread th = new Thread(EndImitation);
    //th.Start();
}

private void button3_Click(object sender, EventArgs e)
{

```

```

        Close();
    }

    public List<string> GetData()
    {
        return DATA;
    }

    private void button4_Click(object sender, EventArgs e)
    {
        button1.Enabled = true;
        listBox6.Items.Add(DateTime.Now.ToString() + ": Начало эксперимента!");
        listBox6.Items.Add(DateTime.Now.ToString() + ": Введите данные!");

        Thread th = new Thread(Imitation1);
        th.Start();

        th = new Thread(Imitation2);
        th.Start();

        th = new Thread(Imitation3);
        th.Start();

        th = new Thread(Imitation4);
        th.Start();
        button4.Enabled = false;
    }

    private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
    {
        if (textBox1.Text == "Введите свой вариант" || textBox1.Text == "Такое значение уже существует")
        {
            textBox1.Text = "";
            textBox1.ForeColor = Color.Black;
        }
    }

    private void textBox1_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.KeyCode != Keys.Enter || !button1.Enabled)
            return;
        if (textBox1.Text == "" || textBox1.Text == "Введите свой вариант" ||
            textBox1.Text == "Такое значение уже существует")
            return;

        if (CheckValue(listBox1) || CheckValue(listBox2) || CheckValue(listBox3) ||
            CheckValue(listBox4) || CheckValue(listBox5))
            return;

        for (int i = 0; i < data_set.Count; i++)
        {
            if (textBox1.Text == data_set[i])
            {
                DATA.Add(data_set[i] + "\n");
                data_set.RemoveAt(i);
            }
        }

        listBox1.Items.Add(textBox1.Text);
    }

```



```

        result_list.Add(DateTime.Now.ToString() + ": " + textBox1.Text);
        Logging("Вы добавили " + textBox1.Text + "!");

        textBox1.Text = "Введите свой вариант";
        textBox1.ForeColor = Color.Gray;
    }
}
}

```

FormMethod3

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Diagnostics;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace diloro
{
    public partial class FormMethod3 : Form
    {
        List<string> data_set = new List<string>();
        String str;
        String big;
        int indexStr, indexLast, indexStart = 0;
        int indexLast2, indexStart2, i = 0;

        private void bigText_TextChanged(object sender, EventArgs e)
        {
        }

        public FormMethod3(List<string> result_list2)
        {
            InitializeComponent();
            data_set = result_list2;
            foreach (String str in data_set)
            {
                bigText.AppendText(str+ " ");
            }
        }

        private void btnSearch_Click(object sender, EventArgs e)
        {
            result.Text = "";
            indexStart = 0;

            if (searchText1.Text != null && searchText1.Text != "")
            {
                str = searchText1.Text;
                big = bigText.Text;
            }
            else

```

