

На правах рукописи *Sign*

Тощев Александр Сергеевич

**Разработка эффективного подхода обработки
производственных задач прикладного характера в
области обслуживания программного обеспечения
и информационной инфраструктуры предприятия**

Специальность 05.13.01 —
«Системный анализ, управление и обработка информации
(информационные технологии)»

Автореферат
диссертации на соискание учёной степени
Кандидат технических наук

Казань — 2015

Работа выполнена в организации Казанский (Приволжский) Федеральный Университет

Научный руководитель: доктор физико-математических наук, профессор
Елизаров А.М.

Официальные оппоненты: **Фамилия Имя Отчество,**
доктор физико-математических наук, профессор,
Не очень длинное название для места работы,
старший научный сотрудник
Таланов Максим Олегович,
кандидат технических наук,
Казанский (Приволжский) Федеральный Университет,
заведующий кафедры робототехники

Ведущая организация: Федеральное государственное бюджетное учреждение науки Институт проблем информатики Российской академии наук

Защита состоится **DD mmmmmmmm YYYY г. в XX часов** на заседании диссертационного совета **NN** на базе **Название учреждения** по адресу: **Адрес.**

С диссертацией можно ознакомиться в библиотеке **Название библиотеки.**

Автореферат разослан **DD mmmmmmmm YYYY** года.

Ученый секретарь
диссертационного совета

NN, д-р физ.-мат. наук

Sign

Фамилия Имя Отчество

Общая характеристика работы

Актуальность темы. В настоящее время в области IT набрали большую популярность системы удаленной поддержки информационной инфраструктуры, так называемый «Аутсорсинг». Ввиду развития рынка компаниям становится невыгодно держать свой штат службы поддержки, и они отдают свою инфраструктуру сторонней компании. Ввиду возросшей интенсивности данного бизнеса возникла большая конкурентность между компаниями на рынке, которая потребовала увеличение эффективности и сокращения издержек, что в свою очередь потребовало системного анализа области и предложения по увеличению эффективности и снижению издержек. В данном контексте рассматривается модель области и модель системы, которая увеличивает эффективность работы в области путем частичной (в некоторых случаях полной) автоматизации обработки инцидентов, начиная с разбора инцидентов на естественном языке и заканчивая поиском решения и применением решения. Главными требованиями к подобной системе являются:

1. Обработка запросов на естественном языке
2. Возможность обучения
3. Общение со специалистом
4. Проведение логических рассуждений: аналогия, дедукция, индукция
5. Умения абстрагировать решение и экстраполировать его на другие решения
6. Способность решить запрос пользователя

На данный момент многие компании ведут разработку подобных систем. Примером такой системы является набирающая популярность система IBM Watson. Подобный класс систем также называют вопросно-ответными системами, например, Wolfram Alpha.

В данной работе была исследована целевая область и построена ее модель. По данной модели была разработана архитектура системы, оптимизирующей эффективность работы. Акцент был сделан на создания мыслящей системы на основе модели мышления Марвина Мински для решения широкого круга проблем, а не специфичных.

Целью данной работы является исследование целевой области, создание ее модели, выработка проблем области, оценка подходов к решению проблем, создание архитектуры и реализация базового прототипа программного комплек-

са обеспечивающего разбор и формализацию входного запроса пользователя и поиск решения данной проблемы.

Для достижения поставленной цели необходимо было решить следующие **задачи**:

1. Провести теоретико-множественный и теоретико-информационный анализ сложных систем в области поддержки информационной инфраструктуры
2. Вычислить технико-экономическую возможность автоматизации целевой области
3. Создать модель целевой области
4. Исследовать модели мышления и выбрать наиболее подходящую
5. На основе выбранной модели мышления разработать модель проблемно-ориентированной системы управления, принятия решений и оптимизации технических объектов в области обслуживания ИТ
6. Создать архитектуру приложения на основе модели
7. Реализовать прототип на основе архитектуры
8. Провести апробацию прототипа на тестовых данных

Основные положения, выносимые на защиту:

1. Теоретико-множественный и теоретико-информационный анализ сложных систем в области поддержки информационной инфраструктуры
2. Модель проблемно-ориентированной системы управления, принятия решений и оптимизации технических объектов в области обслуживания ИТ, ее технико-экономическое обоснование
3. Прототип программной реализации модели проблемно-ориентированной системы управления, принятия решений и оптимизации технических объектов в области обслуживания ИТ
4. Апробация системы на контрольных примерах и ее результаты

Научная новизна:

1. Была создана модель проблемно-ориентированной системы управления, принятия решений и оптимизации технических объектов в области обслуживания ИТ на основе модели мышления Марвина Мински
2. Доказана применимость модели для других областей

3. Была представлена новая модель данных для модели мышления и оригинальный способ хранения модели данных, обеспечивающий быстрый доступ к данным
4. Было выполнено оригинальное исследование модели мышления
5. На основе модели была создана архитектура системы и ее прототип

Практическая значимость Система, разрабатываемая в рамках данной работы носит значимый практический характер. Идея работы зародилась из производственных проблем в IT отрасли, с которыми автор сталкивался каждый день. Только глубокое понимание проблем помогло выбрать правильное решение. Более подробное описание представлено в Главе 1. **Достоверность** полученных результатов обеспечивается результатами выполнения тестов на контрольных примерах. Результаты находятся в соответствии с результатами, полученными другими авторами, экспертными системами и специалистами.

Апробация работы. Основные результаты работы докладывались на:

- RCDL-2014
- AINL-2013
- WCIT-2012
- AMSTA-2015

Личный вклад. Автор принимал активное участие в исследовании целевой области, разработке архитектуры приложения, реализации прототипа, проработки теории, тестировании прототипа.

Публикации. Основные результаты по теме диссертации изложены в 6 печатных изданиях [1], [2], [3], [4], [5], [6], 2 из которых изданы в журналах Scopus, 1 в журнале РИНЦ [5], 4 в тезисах докладов [1], [2], [4], [5], [7].

Содержание работы

Во **введении** обосновывается актуальность исследований, проводимых в рамках данной диссертационной работы, дается общая характеристика работы. **Первая глава** посвящена постановки задачи. Проводится обзор целевой области и обосновывается возможность ее автоматизации. В главе обосновывается состав команд поддержки информационной структуры предприятия. На Диаграмме 1 представлен качественно процентный состав в команд с точки зре-

ния квалификации специалистов.



Рис. 1 — Диаграмма состава команд

В главе приведены результаты анализа категорий проблем, которые решают специалисты [2](#). В главе приведено технико-экономическое обоснование необходимости решения проблемы, где выведен необходимый порог в 50% решения системой целевых инцидентов.

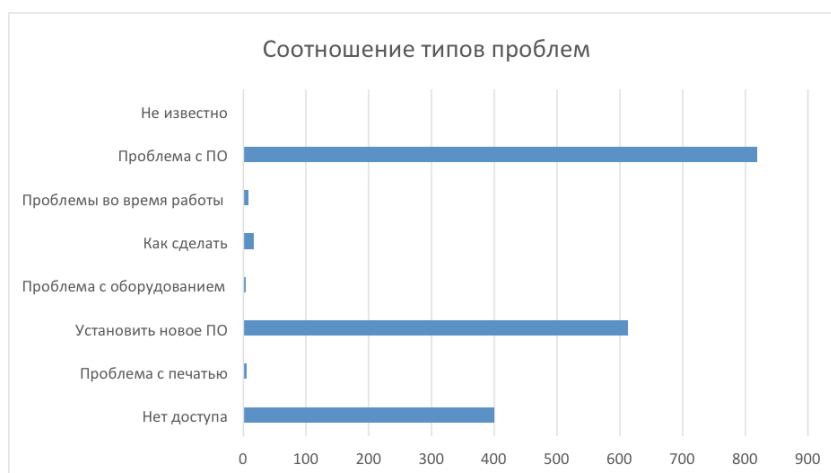


Рис. 2 — Диаграмма соотношений типов проблем

Вторая глава посвящена анализу текущих решений получения, анализа и обработки экспертной информации в области обслуживания программного обеспечения и информационной инфраструктуры. Было выбрано 3 наиболее популярных на сегодняшний день решений: HPOpenView, ServiceNOW, IBMWatson. Были выработаны критерии сравнения и требования к целевой системе. В Таблице 2 приведены результаты сравнения по основным критериям. В главе также был выработан набор тестовых данных, разработаны критерии оценки работы подходов обработки естественного языка, представленных в Таблице 1.

Таблица 1 — Таблица метрик

Метрика	Описание	Формула
Аккуратность	Понимание текста обработчиком	$Ac = \frac{1 - x}{y}$ <p>где x- количество нераспознанных слов, y количество распознанных</p>
Успешно обработанные	Успешно обработанные инциденты	$P = \frac{x}{100}$ <p>где x успешно обработанные</p>
Не успешно обработанные	Неуспешно обработанные инциденты	$N = \frac{y}{100}$ <p>где y неуспешные инциденты</p>
Результативность	Общая результативность обработчика	$R = \frac{P}{N}$
Общий бал	Общая оценка обработчика	$T = Ac + R$

На основе данных критериев был проведен анализ существующих подходов, результаты которого приведены на Диаграмме 3. По итогам главы был сделан вывод, что наиболее эффективен подход, использующийся в комплексе OpenCog Relex.

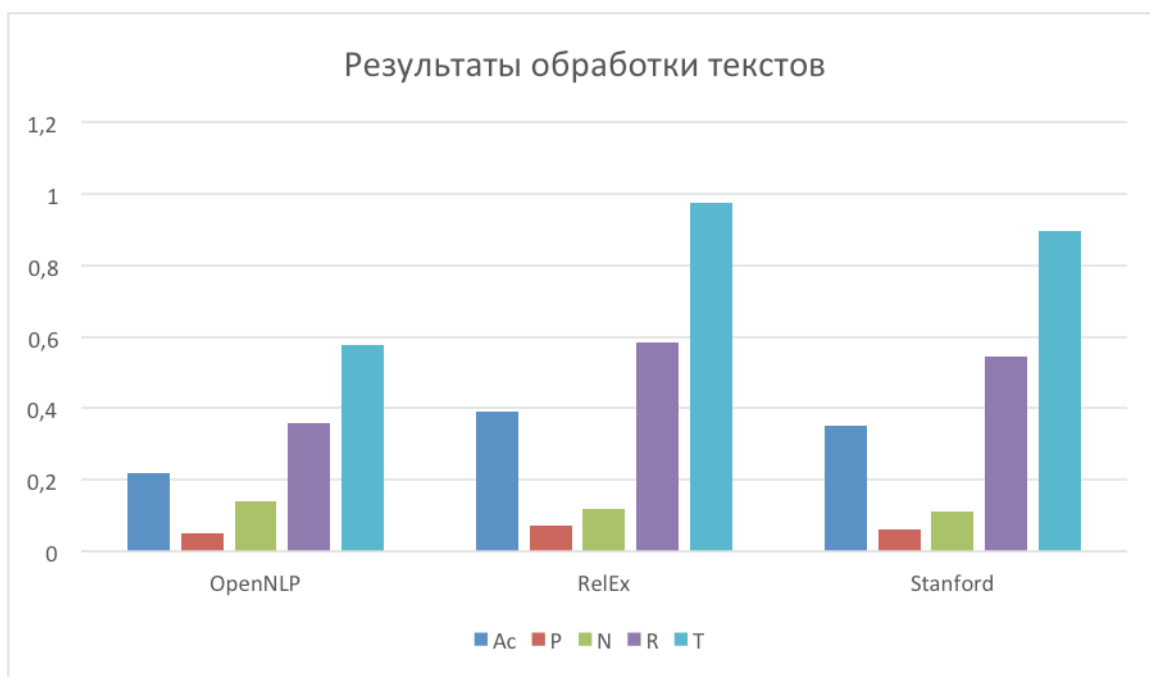


Рис. 3 — Результаты обработки текстов

Таблица 2 — Сравнительный анализ существующих решений

Сравнительный пункт	HP Open View	ServiceNOW	IBM Watson
Мониторинг	Да	Да	Да
Регистрация инцидентов	Да	Да	Да
Управление системами	Да	Нет	Нет
Создание цепи обработки (Workflow) инцидента	Да	Да	Нет
Понимания и формализацию запросов на естественном языке	Нет	Нет	Да
Поиск решений	Нет	Нет	Да
Применение решений	Нет	Нет	Нет
Обучение решению инцидента	Нет	Нет	Да
Умение проводить логические рассуждения: генерализацию, специализацию, синонимичный поиск	Нет	Нет	Нет
Итоговые очки	4	3	5

В третьей главе приведено описание теоретического базиса системы и ее модели. Было рассмотрено четырех различных подхода и модели системы.

- Модель мышления Марвина Мински (TU)
- Модель мышления на базе нейронных сетей
- Модель с использованием Деревьев Принятия Решений (Menta 0.1)
- Модель с использованием Генетических алгоритмов на базе модели мышления Питера Норвига (Menta 0.3)

Модель мышления на базе нейронных сетей Модель на базе нейронных сетей (поддерживающая обучение) была отброшена на предварительной стадии оценки, так как имеет большие требования производительности несовместимые с Технико Экономическим Обоснованием.

Модель с использованием Деревьев Принятия Решений (Menta 0.1)

Данная модель являлась одной из первых, которая была опробована. Модель была основана на деревьях принятия решений. В построение модели данной системы использовались следующие компоненты:

- Обработка запросов на естественном языке
- Поиск решения
- Применение решения

Система была ориентирована на выполнение простых команд, например, добавить поле на форму. Основная функция модели представлена следующим потоком:

1. Получение и формализация запроса
2. Поиск решения при помощи Деревьев Принятия Решений
3. Изменение модели приложения в формате OWL
4. Генерация и компиляция приложения

Основными проблемами данной модели являлось следующее:

1. Отсутствие устойчивости к ошибкам входной информации: грамматическим и содержательным. Например, входной файл не имел отношения к программной системе, модель которой была в базе знаний в формате OWL
2. Система поиска решения работала только в рамках модели одной программы
3. Отсутствовала функция обучения

Модель с использованием Генетических алгоритмов на базе модели мышления Питера Норвига (Menta 0.2-0.3) После работы над ошибками была предпринята попытка сделать поиск решения более универсальным. В рамках данной модели были сформированы основные компоненты будущей модели:

- Критерии Приемки (Acceptance Criteria)
- Формат данных OWL
- Использование логических вычислений

Система содержала внутри себя модель приложения. При помощи генетического алгоритма модель строила из частей новую систему и проверяла ее при помощи логического движка NARS соответствии входным критерия приема. Основными недостатками подхода оказалось:

- Отсутствие обучения
- Отсутствие обработки естественного языка

Кроме того после апробации оказалось, что критерии приема практически описывают необходимое решение (то которое должно быть найдено), что являлось недопустимым.

Модель мышления Марвина Мински (TU) Модель построенная с применением модели мышления Марвина Мински содержала в себе основные концепции предыдущих моделей и показала свою состоятельность на контрольных примерах.

- Acceptance Criteria
- Обучение
- Поиск и применение решения
- Отсутствие обработки естественного языка

Данная модель является более абстрактной и представляется собой верхнеуровневую архитектуру обработки запроса (мышления), где компонентами являются лучшие части предыдущих систем. Крити-Селектор-Путь мышления В 2006 году Марвин Мински опубликовал свою книгу "The emotion machine в которой предложил свой взгляд на систему мышления и памяти человека. В основу теории легла парадигма триплета Критик-Селектор-Путь мышления, k-line для сопоставления знаний. На рисунке 4 представлена схематичное изображение Критика-Селектора-Пути мышления.

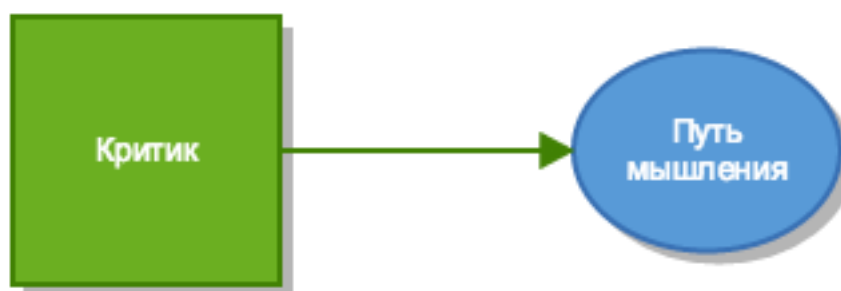


Рис. 4 — Критик-Селектор-Путь мышления

Критик (Critic) представляет собой определенный триггер: внешние обстоятельства, события или иное воздействие. Например, включился свет и зрачки сузились. Обожглись и одернули руку. Критик активируется только когда для этого достаточно обстоятельств. Одновременно могут активироваться несколько критиков. Например, человек решает сложную задачу. Идет активация множество критиков: считать, технические детали, кроме того параллельно может активироваться критик переработки, сообщающей о необходимости отдыха.

Селектор (Selector) занимается выбором определенных ресурсов, которыми также являются Пути мышления.

Путь мышления (WayToThink) это способ решения проблемы. Путь мышления также может активировать следующий критик.

На рисунке 5 представления расширенная модель работы триплета Критик-Селектор-Путь мышления. Критик активирует селектор, который активирует путь мышления (синий круг). Путь мышления в свою очередь может активировать критик или же совершить определенные действия. Например, зажегся зеленый свет светофора, значит можно переходить дорогу.

Если активировалось много критиков, значит проблему нужно уточнить, так как степень неопределенности слишком высока. Если проблема очень похожа, то можно судить по аналогии. В Таблице 3 представлено описание уровней мышления.

Таблица 3 — Описание уровней мышления Марвина Мински

Уровень	Описание
Инстинктивный уровень	На данном уровне происходят инстинктивные реакции (врожденные). Например, боязнь обжечься. Не прыгать под машину. Общую формулу для этого уровня можно выразить как "Если ..., то сделать так".
Уровень обученных реакций	На данной уровне происходит мышление обученных реакций, то есть тех реакций, которыми человек обучается в течение жизни. Например, переходить дорогу на зеленых свет. Общую формулу для этого уровня можно выразить как "Если ..., то сделать так".
Уровень рассуждений	а данной уровне происходит мышление с использованием рассуждений. Если я сделаю так, то будет ... Например, если перебежать дорогу на зеленый свет, то можно успеть вовремя. На данном уровне сравниваются последствия нескольких решений и выбирается оптимальное. Общую формулу для этого уровня можно выразить как "Если ..., то сделать так, тогда будет так".
Рефлексивный уровень	На данном уровне происходит рассуждение с учетом анализа прошлых событий. Например, прошлый раз я побежал на моргающий зеленый и чуть не попал под машину.
Саморефлексивный уровень	На данном уровне происходит оценка себя. Строится определенная модель с помощью которой идет оценка своих поступков. Например, мое решение не пойти на это собрание было неверным, так как я упустил столько возможностей, я был легкомысленный.
Самосознательный уровень	Самозонательный уровень на данный момент характерен только для человека. На данном уровне идет оценка поступков человека с точки зрения высших идеалов и внешних оценок. Например, а что подумают мои друзья? А как бы поступил мой герой?

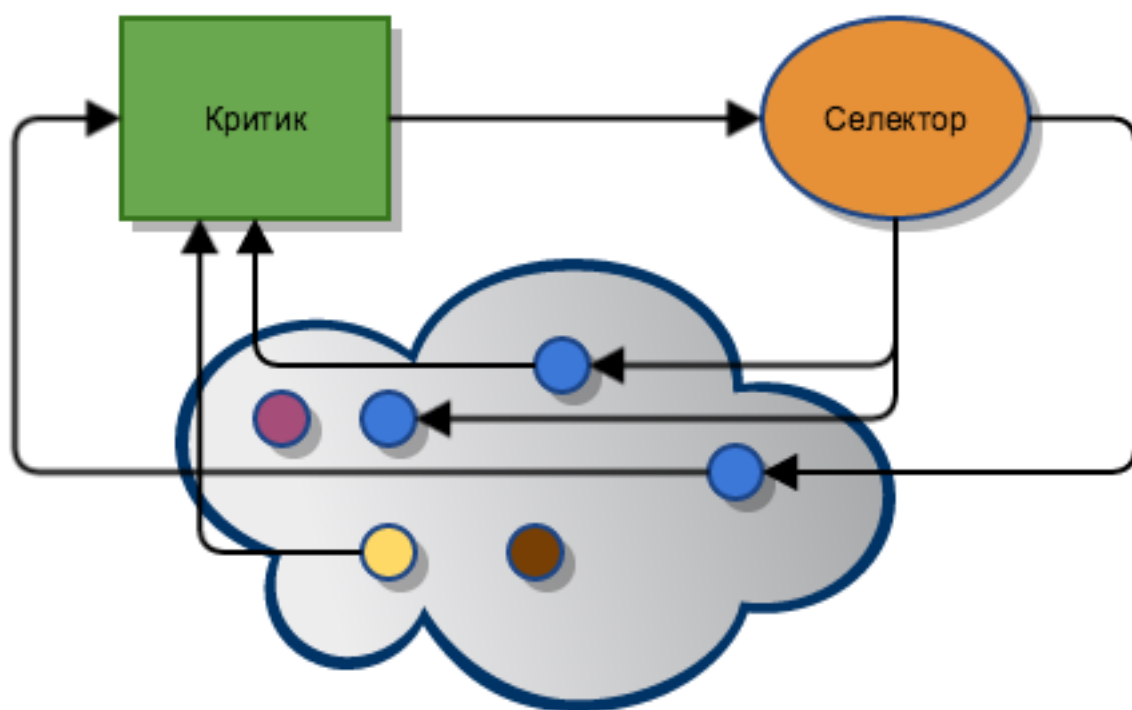


Рис. 5 — Критик-Селектор-Путь мышления в разрезе ресурсов

В **Главе 4** приведены основные результаты работы, которые заключаются в следующем:

- Теоретико-множественный и теоретико-информационный анализ сложных систем в области поддержки информационной инфраструктуры
- Проблемно-ориентированная система управления, принятия решений и оптимизации технических объектов в области обслуживания IT
- Архитектура системы, ее реализация и испытания на модельных данных
- Описание компонентов системы

Архитектура системы представляет собой модульную систему. Основными компоненты системы описаны в Таблице 4. Система может функционировать в режиме обучения и в режиме решения запросов.

Таблица 4 — Основные компоненты системы ThinkingUnderstanding

Компонент	Описание
TU Webservice	Основной компонент взаимодействия со внешними системами, включая пользователя.
CoreService	Ядро системы, содержит основные классы.
DataService	Компонент работы с данными.
Reasoner	Компонент вероятностной логики.
ClientAgent	Компонент выполнения скриптов на целевой машине.
MessageBus	Шина данных для системы.

В главе приводятся основной поток работы приложения.

1. Поступает запрос от пользователя
User had received wrong application. User has ordered Wordfinder Business Economical. However she received wrong version, she received Wordfinder Tehcnical instead of Business Economical. Please assist.
2. GoalManger устанавливает цель системы HelpUser
3. Активируется набор Critic, привязанный к данной цели
4. PreliminaryAnnorator разбирает фразу

5. KnowledgeBaseAnnotator создает семантическую сеть и ссылки на нее
6. Critic на Рефлексивном уровне запускает WayToThink ProblemSolving с целью: ResolveIncident
7. Critic на Рефлексивном уровне выбирает WayToThink KnowingHow
 - (a) Запускаются параллельно все Critic, которые привязаны к IncidentClassification Critic, который привязан к ResolveIncident цели, в данном случае это DirectInstruction, ProblemWithDesiredState, ProblemWithoutDesiredState ??
 - (b) Selector выбирает наиболее вероятный результат работы среди всех результатов компонентов. В данном случае будет результат работы Problem Description with desired state.
 - (c) KnowingHow сохраняет варианты выбора Selector.
 - (d) Simulation WayToThink с параметрами Создать модель текущий ситуации создает модель CurrentSituation. User, Software
 - (e) Reformulation WayToThink, используя результаты предыдущего шага синтезирует артефакты, которых не хватает, чтобы получить CurrentState и DesiredState. DesiredState не указан явно. WayToThink запускает Critic размышления, чтобы найти корень проблемы. Critic размышления находит CurrentState- Wordfinder Tehcnical, DesiredState- Wordfinder Business Economical
 - (f) Рефлексивные Critic оценивают состояние системы - на каком шаге она находится, и если цель не достигнута, то запускают другой WayToThink, который был возвращен, например, DirectInstruction.
 - (g) Critic генерации решения запускает KnowingHow ?? WayToThink, ExtensiveSearch.
 - (h) Selector выбирает наиболее вероятный путь мышления. В данном случае ExtensiveSearch, который будет находить решения, позволяющие привести систему в необходимое состояние (DesiredState). Если он не сможет, то он инициирует коммуникацию с пользователем.

8. Рефлексивный Critic проверяет состояние системы. Если Цель достигнута, то пользователю посылается ответ.
9. Само Сознательные Critic активируется на данном шаге и сохраняют информацию о затратах на решение.

В главе представлены методика и результаты верификации работы. В **заключении** приведены основные выводы по работе, которые заключаются в следующем:

Основные результаты работы заключаются в следующем.

1. На основе анализа предметной области (поддержка информационной структуры предприятия) была выявлена потребность и возможность в автоматизации. Была построена модель предметной области. На основе модели предметной области, модели Марвина Мински была разработана модель проблемно-ориентированной системы принятия решений в области поддержки информационной структуры предприятия.
2. Испытания комплекса на модельных данных показали работоспособность модели и архитектуры.
3. Для выполнения поставленных задач был создан программный комплекс обработки, решения инцидентов и обучения на естественном языке.

Представленная в данной работе модель мышления, ее архитектура и реализация является уникальной в своем роде. На момент написания это была единственная реализация модели мышления Марвина Мински.

В работе были проведены исследования согласно паспорту специальности 05.13.01, сопоставление приведено в Таблице 5.

Таблица 5 — Сопоставление направлений исследования специальности 05.13.01 и исследований, проведенных в работе

Направление исследования	Результат работы
Продолжение следует	

Таблица 5 – продолжение

Направление исследования	Результат работы
Разработка критериев и моделей описания и оценки эффективности решения задач системного анализа, оптимизации, управления, принятия решений и обработки информации	В рамках работы была разработана модель системы принятия решения и обработки информации в области решения запросов пользователя на естественном языке.
Разработка проблемно-ориентированных систем управления, принятия решений и оптимизации технических объектов	По модели, разработанной в предыдущем пункте был разработан прототип системы принятия решения Thinking-Understanding, который был испытан на модельных данных.
Методы получения, анализа и обработки экспертной информации	В рамках системы TU был разработан метод обработки экспертной информации - обучение при помощи модели мышления TU, основанной на принципах модели мышления Марвина Мински.
Разработка специального математического и алгоритмического обеспечения систем анализа, оптимизации, управления, принятия решений и обработки информации	В рамках разработки системы TU были разработаны специальные алгоритма для анализа запросов пользователя и принятия решений.
Разработка специального математического и алгоритмического обеспечения систем анализа, оптимизации, управления, принятия решений и обработки информации	В рамках разработки системы TU были разработаны специальные алгоритма для анализа запросов пользователя и принятия решений.
Продолжение следует	

Таблица 5 – продолжение

Направление исследования	Результат работы
Теоретико-множественный и теоретико-информационный анализ сложных систем	В рамках работы был проведен комплексный анализ области поддержки программного обеспечения, с помощью которого была построена система данной области и выделены участки для оптимизации принятия решений.
Методы и алгоритмы интеллектуальной поддержки при принятии управленческих решений в технических системах	Система, разработанная в рамках данной работы в включает в себя инновационные методы и алгоритмы поддержки принятия решений, использующих в своей основе модель мышления на базе модели мышления Человека, описанной в книге Марвина Мински.
Визуализация, трансформация и анализ информации на основе компьютерных методов обработки информации	В Главе 1 представлена наглядная визуализация данных по системному анализу области удаленной поддержки инфраструктуры.

Разработанная в рамках работы системы не является узко-специализированной. Она также подходит для других областей, где требуется поддержка принятия решений. Например, при постановке медицинского диагноза, чтобы отбросить ложные диагнозы.

Например, систему можно обучить органам человека и их взаимосвязи. Далее можно обучить каким заболеваниям подвержен тот или иной орган. Далее к каждому заболеванию добавить симптом. После этого можно делать запрос с симптомами и система выдаст список вероятных заболеваний с их вероятностью и способы их лечения.

В области диагностики проблем в машиностроение. Обучить систему узлам автомобиля, проблемам с ними связанными, признаками этих проблем и способами их устранения.

Работа над системой продолжается. Система принимает участия в нескольких грантах.

Публикации автора по теме диссертации

1. Тоцев А. С. К новой концепции автоматизации программного обеспечения // *Труды Математического центра имени Н.И. Лобачевского. Материалы Десятой молодежной научной школы-конференции 'Лобачевские чтения - 2011. - Казань, 31 октября - 4 ноября 2011'*. — 2011. — Vol. 44. — 2 pp.
2. Toshchev A. Talanov M. Krehov A. Khasianov A. Thinking-Understanding approach in IT maintenance domain automation // *Global Journal on Technology, Vol 3 (2013): 3rd World Conference on Information Technology (WCIT-2012)*. — 2013. — Т. 3. — Режим доступа: <http://www.world-education-center.org/index.php/P-ITCS/issue/view/96>.
3. Toshchev A. Thinking model and machine understanding in automated user request processing // *CEUR Workshop Proceedings*. — 2014. — Т. 1297.
4. Toshchev A. Talanov M. Thinking model and machine understanding of English primitive texts and it's application in Infrastructure as Service domain // *Proceedings of AINL-2013*. — 2013. — Режим доступа: <http://ainlconf.ru/material201303>.
5. Toshchev A. Talanov M. ARCHITECTURE AND REALIZATION OF INTELLECTUAL AGENT FOR AUTOMATIC INCIDENT PROCESSING USING THE ARTIFICIAL INTELLIGENCE AND SEMANTIC NETWORKS // *Ученые записки ИСГЗ 2078-6980*. — 2014. — Т. 2. — Режим доступа: <http://ainlconf.ru/material201303>.
6. Toshchev A. Talanov M. Thinking Lifecycle as an Implementation of Machine Understanding in Software Maintenance Automation Domain // *Agent and Multi-Agent Systems: Technologies and Applications: 9th KES International Conference, KES-AMSTA 2015 Sorrento, Italy, June 2015, Proceedings (Smart Innovation, Systems and Technologies)*. — 2015.
7. Toshchev A. Talanov M. Computational emotional thinking and virtual neurotransmitters // *International Journal of Synthetic Emotions (IJSE)*. — 2014. — Т. 06/2014.