

На правах рукописи



**Тощев Александр Сергеевич**

**ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА  
ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ  
ИТ-СЛУЖБЫ ПРЕДПРИЯТИЯ**

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

**Автореферат**  
диссертации на соискание учёной степени  
кандидата физико-математических наук

Казань — 2016

Работа выполнена в Казанском (Приволжском) федеральном университете

Научный руководитель: доктор физико-математических наук, профессор, заслуженный деятель науки Республики Татарстан  
**А.М. Елизаров**

Официальные оппоненты: **Соловьев Валерий Дмитриевич**,  
доктор физико-математических наук, профессор,  
Казанский (Приволжский) федеральный университет,  
Институт филологии и межкультурной коммуникации им. Льва Толстого,  
ведущий научный сотрудник

**Фамилия Имя Отчество**,  
кандидат физико-математических наук,  
Основное место работы с длинным длинным длинным  
длинным названием,  
старший научный сотрудник

Ведущая организация: Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования с длинным длинным длинным длинным названием

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**

**Целью** работы является разработка интеллектуальной системы повышения эффективности деятельности ИТ-службы (ИТ — информационные технологии) предприятия.

**Область исследования** — разработка методов и алгоритмов решения задач системного анализа, оптимизации, управления, принятия решений и обработки информации в ИТ-отрасли.

**Предметом исследования** является процесс регистрации и устранения проблемных ситуаций, возникающих в ИТ-инфраструктуре предприятия.

**Методы исследования** — экспериментальные методы: метод наблюдений, проведение экспериментов; теоретические методы: метод идеализации, метод формализации; специальные методы: системное моделирование, системный анализ.

Для достижения поставленной цели были решены следующие **задачи**:

1. Провести теоретико-множественный и теоретико-информационный анализ сложных систем в области поддержки информационной инфраструктуры;
2. Разработать модель проблемно-ориентированной системы управления, принятия решений и оптимизации процесса принятия, анализа и обработки запросов пользователей в области обслуживания информационной структуры предприятия;
3. На основе построенной модели разработать архитектуру и создать прототип интеллектуальной вопросно-ответной системы повышения эффективности деятельности ИТ-службы предприятия;
4. Провести апробацию прототипа на тестовых данных.

**Основные положения, выносимые на защиту:**

1. Результаты теоретико-множественного и теоретико-информационного анализа сложных систем в области поддержки ИТ-инфраструктуры предприятия;
2. Построенная модель проблемно-ориентированной системы управления, принятия решений и оптимизации процессов обработки за-

просов пользователей в области обслуживания ИТ-инфраструктуры предприятия;

3. Созданный прототип программной реализации модели проблемно-ориентированной системы управления, принятия решений и оптимизации обработки запросов пользователей в области обслуживания ИТ-инфраструктуры предприятия;
4. Результаты апробации прототипа проблемно-ориентированной системы управления, принятия решений и оптимизации деятельности на контрольных примерах и анализ ее результатов.

**Научная новизна** проведенного исследования состоит в следующем:

1. На основе модели мышления создана модель проблемно-ориентированной системы управления, принятия решений в области обслуживания ИТ-инфраструктуры предприятия;
2. Представлены новая модель данных для модели мышления и оригинальный способ хранения данных для этой модели, эффективный по сравнению со стандартными способами хранения данных (такие как реляционные базы данных);
3. Выполнено оригинальное исследование моделей мышления применительно к области обслуживания информационной структуры предприятия;
4. На основе модели мышления Мински созданы архитектура системы обслуживания информационной структуры предприятия и программный прототип этой системы.

**Практическая значимость.** Система, разработанная в рамках данной диссертации, носит значимый практический характер. Идея работы зародилась под влиянием производственных проблем в ИТ-отрасли, с которыми автор сталкивался каждый день в процессе разрешения различных инцидентов, возникающих в деятельности службы технической поддержки ОАО «АйСиЭл КПО-ВС (г. Казань)» — одном из крупнейших системообразующих предприятий ИТ-отрасли Республики Татарстан. Поэтому было необходимо выработать глубокое понимание конкретной предметной области, чтобы выбрать приемлемое решение, получившее практическое применение при реализации проекта информационной поддержки ИТ-инфраструктуры конкретного предприятия.

**Достоверность** полученных научных результатов и выработанных практических рекомендаций базируется на корректной постановке общих и частных рассматриваемых задач, использовании известных фундаментальных теоретических положений системного анализа, достаточном объёме данных, использованных при статистическом моделировании, и широком экспериментальном материале, использованном для численных оценок достижимых качественных показателей.

Исследования, проведенные в диссертации, соответствуют паспорту специальности 05.13.01 — Системный анализ, управление и обработка информации, сопоставление приведено в таблице 1.

Таблица 1 — Сопоставление направлений исследований в рамках специальности 05.13.01 и исследований, проведенных в диссертации

Направление исследования	Результат работы
Разработка критериев и моделей описания и оценки эффективности решения задач системного анализа, оптимизации, управления, принятия решений и обработки информации	Разработана модель системы принятия решения и обработки информации в сфере поддержки ИТ-инфраструктуры предприятия.
Разработка проблемно-ориентированных систем управления, принятия решений и оптимизации технических объектов	Разработан прототип Thinking Understanding (TU) системы принятия решений в сфере поддержки ИТ-инфраструктуры предприятия, который был испытан на модельных данных.
Методы получения, анализа и обработки экспертной информации	Разработан метод обработки экспертной информации с возможностью обучения при помощи прототипа TU.
Разработка специального математического и алгоритмического обеспечения систем анализа, оптимизации, управления, принятия решений и обработки информации	Созданы специальные алгоритмы для анализа запросов пользователей и принятия решений.

**Таблица 1 – продолжение**

Направление исследования	Результат работы
Теоретико-множественный и теоретико-информационный анализ сложных систем	Проведен комплексный анализ области поддержки программного обеспечения крупного ИТ-предприятия, с помощью которого построена модель области и выделены направления и возможности оптимизации принятия решений.

**Апробация работы.** Основные результаты диссертационной работы докладывались на следующих конференциях:

- Десятая молодежная научная школа-конференция «Лобачевские чтения —2011». Казань, 31 октября – 4 ноября 2011 года;
- Международная конференция "3rd World Conference on Information Technology (WCIT-2012)". Barcelona, 14 – 16 November 2012, Spain;
- II Международная конференция «Искусственный интеллект и естественный язык (AINL-2013)». Санкт-Петербург, 17 – 18 мая 2013 года;
- VI Международная научно-практическая конференция «Электронная Казань 2014». Казань, 22 – 24 апреля 2014 года;
- XVI Всероссийская научная конференция «Электронные библиотеки: перспективные методы и технологии, электронные коллекции (RCDL-2014)». Дубна, 13 – 16 октября 2014 года;
- Семинары по программной инженерии "All-Kazan Software Engineering Seminar (AKSES-2015)". Kazan, 9 April 2015;
- Международная конференция "Agents and multi-agent systems: technologies and applications (AMSTA-2015)". Sorrento, 17 – 19 June 2015, Italy.

Практическая апробация результатов работы проводилась на выгрузке инцидентов из системы регистрации запросов службы технической поддержки ИТ-инфраструктуры ОАО «АйСиЭл КПО-ВС (г. Казань)». Созданная система показала требуемые результаты обработки данной информации (процент успешно обработанных запросов составил более чем 30%).

**Личный вклад.** Автор исследовал целевую область: провел анализ запросов пользователей и классифицировал их; построил модель целевой области

и выявил возможности оптимизации. Совместно с Талановым Максимом Олеговичем создал базовую архитектуру системы. Автор разработал компоненты системы, провел испытание системы на экспериментальных данных и отладил ее работу.

**Публикации.** Основные результаты по теме диссертации изложены в 11 печатных изданиях [1–11], из которых статьи [7; 8] проиндексированы в БД Scopus и входят в перечень ВАК, статья [8] проиндексирована в БД Web of Science и входит в перечень ВАК, работа [9] опубликована в журнале из списка ВАК, статья [4] проиндексирована в БД РИНЦ, работы [1], [2–4] опубликованы в материалах международных и всероссийских конференций.

## Содержание работы

Во **введении** обоснована актуальность исследований, проводимых в рамках диссертации, дана общая характеристика работы. Проведен обзор области обслуживания информационной структуры предприятия и дана постановка задач.

**Первая глава** посвящена обзору интеллектуальных систем регистрации и анализа проблемных ситуаций, возникающих в ИТ-инфраструктуре предприятия. Здесь представлен сравнительный анализ систем регистрации и устранения проблемных ситуаций; определены основные требования к интеллектуальным системам регистрации и анализа проблемных ситуаций в ИТ-сфере. Одним из важных элементов подобных систем является обработка естественного языка, поэтому здесь представлен сравнительный анализ методов и программных комплексов обработки текстов.

При проведении анализа были использованы следующие средства обработки естественного языка: Open NLP (проект [opennlp.apache.org](http://opennlp.apache.org)); Relex (проект [opencog.org](http://opencog.org)); StanfordParser (проект <http://nlp.stanford.edu/software/lex-parser.shtml>). Оценка качества функционирования этих средств проводилась при помощи метрик, представленных в таблице 2, а полученные результаты приведены на рисунке 1. Как видно, максимальный результат по всем трем метрикам показывает система Relex, она и была выбрана в качестве основного средства обработки естественного языка.

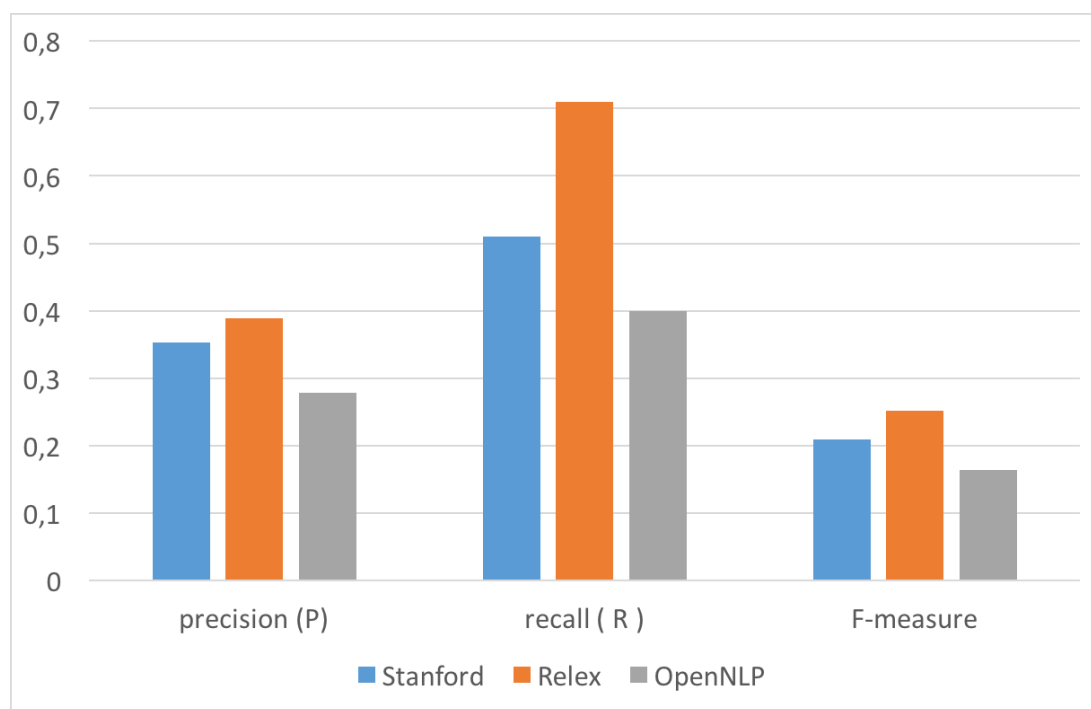


Рис. 1 — Результаты анализа средств обработки естественного языка

Таблица 2 — Таблица метрик

Метрика	Описание	Формула
Precision	Точность	$P = \frac{tp}{tp + fp},$ <p>где P — precision, tp — успешно обработанные слова, fp — ложно успешные</p>
Recall	Чувствительность	$R = \frac{tp}{tp + fn},$ <p>где R — recall, tp — успешно обработанные слова, fn — ложно неуспешные</p>
F	F — measure (результативность)	$F = \frac{P * R}{P + R},$ <p>где P — precision, R — recall.</p>



Кроме того, в главе 1 проведен анализ существующих программных систем автоматизации в области поддержки информационной структуры предприятия: HP Open View<sup>1</sup>, ServiceNOW<sup>2</sup>, IBM Watson<sup>3</sup>. Все рассмотренные системы не соответствуют полному набору необходимых требований. В таблице 3 приведены сводные данные по системам — наличие той или иной функции у рассматриваемой системы. Как видно, ни одно из рассмотренных решений не способно проводить логические рассуждения. Наиболее развитой на сегодняшний день программной системой является комплекс IBM Watson.

Таблица 3 — Сравнительный анализ существующих программных систем.

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
Понимание и формализация запросов на естественном языке	Нет	Нет	Да
Поиск решений	Нет	Нет	Да
Применение решений	Нет	Нет	Нет
Обучение	Нет	Нет	Да
Умение проводить логические рассуждения: генерализацию, специализацию, синонимичный поиск	Нет	Нет	Нет

**Вторая глава** посвящена построению модели интеллектуальной системы принятия решений для регистрации и анализа проблемных ситуаций в ИТ-инфраструктуре предприятия. Рассмотрены три принципиальных подхода к решению проблемы.

<sup>1</sup>[https://ru.wikipedia.org/wiki/HP\\_OpenView](https://ru.wikipedia.org/wiki/HP_OpenView)

<sup>2</sup><http://www.servicenow.com/>

<sup>3</sup><http://www.ibm.com/smarterplanet/us/en/ibmwatson/>

- модель Menta 0.1, построенная с использованием деревьев принятия решений;
- модель Menta 0.3, построенная с использованием генетических алгоритмов;
- модель TU 1.0, основанная на модели мышления Марвина Мински.

Необходимо отметить, что модель, построенная на базе нейронных сетей (поддерживающая обучение), была отброшена на предварительной стадии оценки, так как она предъявляет большие требования к производительности, что в свою очередь порождает высокую стоимость. Далее каждая модель будет описана подробно.

**Модель Menta 0.1, построенная с использованием деревьев принятия решений**, являлась одной из первых, которая была опробована. При построении модели использовались следующие компоненты: обработка запросов на естественном языке; поиск решения; применение решения.

Система ориентирована на выполнение простых команд, например, добавить поле «в форму». В целом работа системы описывается следующим алгоритмом:

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.

В результате проведения экспериментов было выявлено 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; система поиска решения работала только в рамках модели одной программы; отсутствовала функция обучения.

**Модель Menta 0.3, построенная с использованием генетических алгоритмов.** В данную модель по сравнению с предыдущей были добавлены модуль логики для оценки решения и модуль генетических алгоритмов для генерации решения. В рамках модели Menta 0.3 были отработаны следующие основные компоненты будущей итоговой модели: критерии приемки (Acceptance Criteria); How-To — для хранения решений проанализированных проблем; формат данных OWL; использование логических вычислений для проверки реше-

ния. Система Menta 0.3 содержала внутри себя модель целевого приложения (как и Menta 0.1) и список решений тех или иных проблем (How-To). При помощи генетического алгоритма модель строила решение, проверяла его при помощи логического движка NARS<sup>4</sup> на соответствие входным критериям приемки. С точки зрения генетических алгоритмов, это — функция отбора особей из поколения.

В результате проведения экспериментов были выявлены следующие проблемы: отсутствие обучения; отсутствие обработки естественного языка; после апробации оказалось, что критерии приемки практически описывают необходимое решение (то, которое должно быть найдено), что являлось недопустимым.

**Модель TU 1.0, основанная на модели мышления Марвина Мински,** была построена с применением известной теории Марвина Мински<sup>5</sup> сохранила следующие основные концептуальные элементы предыдущих моделей и показала свою состоятельность на контрольных примерах: Acceptance Criteria; обучение; поиск и применение решения; отсутствие обработки естественного языка. Данная модель является более универсальной и представляет собой верхнеуровневую архитектуру обработки запроса (мышления), где компонентами являются лучшие части предыдущих систем. Реализация модели получила название TU.

Одним из основных компонентов системы является Критик – Селектор – Образ мышления(далее  $T^3$ ), схематичное изображение которого представлено на рисунке 2. Критик реагирует, Селектор выбирает ресурс, Образ мышления выполняет работу.



Рис. 2 —  $T^3$

<sup>4</sup><http://www.cogsci.indiana.edu/farg/peiwang/papers.html>

<sup>5</sup>[https://en.wikipedia.org/wiki/The\\_Emotion\\_Machine](https://en.wikipedia.org/wiki/The_Emotion_Machine)

*Критик (Critic)* представляет собой определенный переключатель: внешние обстоятельства, события или иное воздействие. Например, «включился свет, и зрачки сузились», «обожглись и одернули руку». Критик активируется только тогда, когда для этого достаточно обстоятельств. Одновременно может активироваться несколько критиков. Например, человек решает сложную задачу, идет активация множество критиков: выполнить расчет, уточнить технические детали. Кроме того, параллельно может активироваться критик контроля уровня загруженности, сообщающий о необходимости отдыха.

*Селектор (Selector)* занимается выбором необходимым ресурсов, которыми, например, могут быть: Критик, Образ мышления.

*Образ мышления (WayToThink)* — это способ решения проблемы. Он может быть сложным и, например, может активировать других критиков. Так, размышляя над проблемой, специалист понимает, что нужно произвести полный перебор, и тут он решает поискать готовое решение: а может кто-то уже сделал такой перебор и можно будет его использовать. Здесь «поиск готового решения» является критиком внутри образа мышления «поиск решения».

На рисунке 3 представлена расширенная модель работы  $T^3$ . Критик активирует Селектор, который возвращает ресурс Образ мышления (кругами отмечены различные ресурсы: Критики, Селекторы, Образы мышления и т. д.). Последний в свою очередь может активировать нового Критика или же совершить определенные действия. Например, зажегся зеленый свет светофора, значит, можно переходить дорогу. Под ресурсами здесь понимается набор знаний из базы знаний: Критики, Селекторы, Образы мышления, готовые решения.

Если активировалось много Критиков, то проблему нужно уточнить, так как степень неопределенности слишком высока. Если проблема очень похожа на уже проанализированную, то можно действовать и судить по аналогии.

Другой важной концепцией теории Мински являются уровни мышления. Эта концепция распределяет активность мышления между 6-ю уровнями: чем выше уровень, тем сильнее активность. В Таблице 4 представлено описание уровней мышления с примерами.

На этом исследование моделей мышлений было завершено и были сделаны выводы.

Для программной экспертной системы очень важно обладать способностью мыслить и рассуждать, например, действовать по аналогии. Множе-

Таблица 4 — Описание шести уровней мышления, заложенных в модель Мински

Уровень	Описание
Инстинктивный уровень	Происходят инстинктивные реакции (врожденные). Например, коленный рефлекс. Общую формулу для этого уровня можно выразить как «если ..., то сделать так».
Уровень обученных реакций	Используются накопленные знания, то есть те знания, которым человек обучается в течение жизни. Например, переходить дорогу на зеленый свет. Общую формулу для этого уровня можно описать как «если ..., то сделать так».
Уровень рассуждений	Мышление с использованием рассуждений. 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	Рассуждения с учетом анализа прошлых событий. Например, «в прошлый раз я побежал на моргающий зеленый и чуть не попал под машину».
Саморефлексивный уровень	Построение определенной модели, 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 легкомысленным».
Самосознательный уровень	Оценка своих поступков с точки зрения высших идеалов и оценок окружающих. Например, «а что подумают мои друзья? А как бы поступил мой герой?»

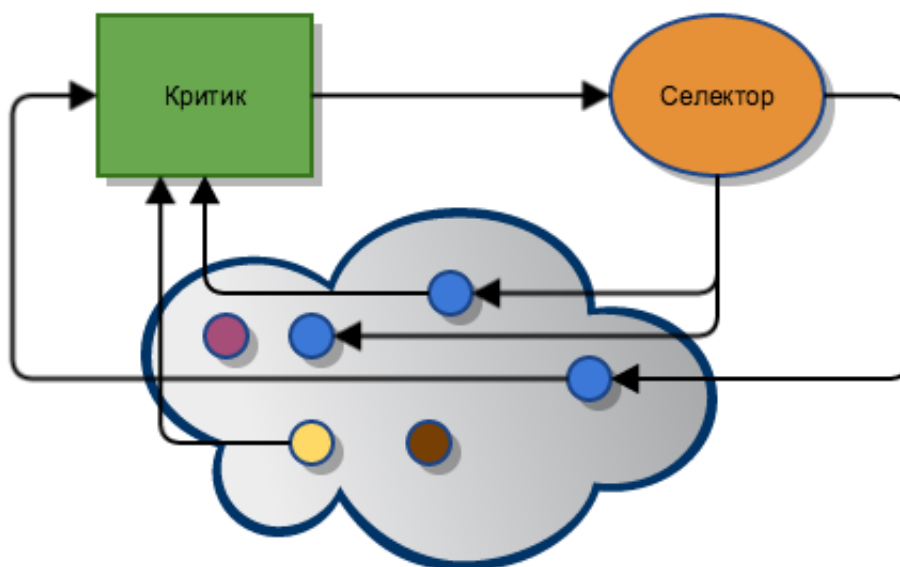


Рис. 3 —  $T^3$  в разрезе ресурсов

ство запросов типично и отличается лишь параметрами. Например, пожалуйста, установите Office, Antivirus и т. д. Также для экспертной системы важно уметь абстрагировать специализированные рецепты решения. Например, система научилась разрешать инцидент "Please install Firefox", абстрагировав данный инцидент до степени "Please install browser", система сможет тем же способом устранить проблему "Please install Chrome", так как концепции "Firefox" и "Chrome" связаны через концепцию "Browser".

После рассмотрения нескольких моделей была выбрана модель мышления Марвина Мински, так как она наиболее точно ложится на целевую область поддержки ИТ-инфраструктуре предприятия. На основе подхода Мински была построена модель системы, которая реализует основные функции: обучение, понимание инцидента, поиск решения, применение решения.

В **третьей главе** приведено описание архитектуры и реализации системы, основанной на модели Thinking Understanding (TU). Архитектура представляет собой модули. Основные компоненты системы описаны в Таблице 5. Система может функционировать в режиме обучения и в режиме устранения проблемных ситуаций.

Таблица 5 — Основные компоненты системы Thinking Understanding

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 3 приведено детальное описание всех компонентов и подкомпонентов. Для лучшего понимания представлены описание механизма взаимодействия компонентов и общий сценарий использования системы.

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." («Пользователь получил неверное приложение. Пользователь заказал приложение "Wordfinder. Бизнес версия", но получил неверную версию, — "Wordfinder. Техническая версия". Пожалуйста, помогите»);
2. Компонент GoalManger (Менеджер целей) устанавливает цель системы HelpUser (Помочь пользователю);
3. Главный компонент Thinking Life Cycle (далее TLC) активирует набор компонент Critic (Критик), привязанный к данной цели (HelpUser);
4. Активируется компонент PreliminaryAnnorator (Предварительный обработчик), который разбирает запрос, проводя орфографическую коррекцию и предварительный разбор;
5. Компонент KnowledgeBaseAnnotator (разбор при помощи накопленных знаний) создает семантическую сеть и ссылки на нее;
6. Компонент Critic (Критик), привязанный к цели HelpUser на Рефлексивном уровне, запускает WayToThink (Образ мышления)

ProblemSolving (Разрешить проблемную ситуацию) с целью: ResolveIncident;

7. Компонент Critic на Рефлексивном уровне выбирает WayToThink KnowingHow (Поиск рецепта решения);

- (a) Запускаются параллельно все компоненты класса Critic, которые привязаны к цели ResolveIncident (Решить проблему), в данном случае это DirectInstruction (прямые инструкции), ProblemWithDesiredState (проблемы с желаемым состоянием), ProblemWithoutDesiredState (проблема без желаемого состояния);
- (b) Компонент Selector (Селектор) выбирает среди всех результатов наиболее вероятный результат работы. В данном случае им будет Problem Description with desired state (Проблема с желаемым состоянием);
- (c) Компонент KnowingHow сохраняет варианты выбора Selector;
- (d) Компонент Simulation (Моделирование) WayToThink с параметрами «создать модель текущий ситуации» создает: концепцию существующей ситуации (CurrentState), концепцию пользователя, концепцию программного обеспечения;
- (e) Компонент Reformulation WayToThink (Компонент дополнения), используя результаты предыдущего шага, синтезирует артефакты, которых не хватает, чтобы получить из CurrentState DesiredState (Желаемое состояние), так как он не указан явно. WayToThink запускает Critic размышления, чтобы найти корень проблемы. Он находит CurrentState (настоящее состояние) — Wordfinder Technical и DesiredState (состояние, которое нужно пользователю) — Wordfinder Business Economical;
- (f) Рефлексивные Critic оценивают состояние системы — на каком шаге она находится, и если цель не достигнута, то запускают другой WayToThink, например, DirectInstruction;



- (g) Компонент Critic Solution Generator (Компонент генерации решения) запускает KnowingHow WayToThink, ExtensiveSearch (Поиск решения);
  - (h) Компонент Selector выбирает наиболее вероятный образ мышления. В данном случае это будет ExtensiveSearch, который будет находить решения, позволяющие привести систему в необходимое пользователю состояние (DesiredState), если сделать это невозможно, то система инициирует коммуникацию с пользователем.
8. Рефлексивный Critic проверяет состояние системы. Если Цель достигнута, то пользователю посылается ответ.
  9. На данном шаге активируются компоненты класса Critic на самосознательном уровне, которые сохраняют информацию о затратах на решение.

Для работы системы создана уникальная модель данных — TU Knowledge, которая сочетает в себе OWL и графовую базу данных. Язык OWL, появившейся для структурирования информации в Вебе, обрел широкое использование во многих схемах данных, так как дал возможность дополнительного расширенного описания взаимосвязи между данными.

В главе 4 приведены результаты экспериментальных исследований эффективности работы модели TU. Система показала свою жизнеспособность на контрольных примерах. Были проведены тесты в сравнении с работой специалиста. Был выбран контрольный список инцидентов. Сравнялся поиск решения для инцидентов. Основное время при опросе специалиста тратилось на коммуникацию. В таблице 6 приведены результаты сравнения. Тесты были выполнены на компьютере Intel Core i7 1700 MHz, 8GB RAM, 256 GB SSD, FreeBSD. Из результатов видно, что система работает так же или лучше, чем специалист технической поддержки.

Таблица 6 — Результаты сравнения с работой специалиста технической поддержки

Инцидент	TSS1 (.мс)	TU (.мс)
Tense is kind of concept (Время — это концепция).	15000	385
Please install Firefox (Установите Firefox).	9000	859
Browser is an object (Браузер — это объект).	20000	400

**Таблица 6 – продолжение**

<b>Инцидент</b>	<b>TSS1 (.мс)</b>	<b>TU (.мс)</b>
Firefox is a browser (Firefox — это браузер).	5000	659
Install is an action (Установить — это действие).	8000	486
User miss Internet Explorer 8 (У пользователя нет Internet Explorer 8).	10000	10589
User needs document portal update (Пользователю требуется обновление документов).	15000	16543
Add new alias Host name on host that alias is wanted to: hrportal.lalala.biz IP address on host that alias is wanted to: 322.223.333.22 Wanted Alias: webadviser.lalala.net (Добавьте, пожалуйста, новую ссылку на hrportal.lalala.biz через 322.223.333.22).	10000	18432
Outlook Web Access (CCC) — 403 — Forbidden: Access is denied (Нет доступа к Outlook Web Access (CCC)).	15000	10342
PP2C — Cisco IP communicator. Please see if you can fix the problem with the ip phone, it's stuck on configuring ip + sometimes Server error rejected: Security etc (PP2C — коммуникатор Cisco IP. Пожалуйста, помогите исправить проблему с ИП-телефоном, он застревает во время конфигурирования и иногда показывает ошибку «Безопасность»).	13000	12343

**В заключении** диссертации приведены основные выводы по работе:

Решены следующие задачи и достигнуты следующие результаты.

1. Создана модель проблемно-ориентированной системы управления, принятия решений в области обслуживания информационной структуры предприятия на основе модели мышления;
2. Представлены новая модель данных для модели мышления и оригинальный способ ее хранения, более эффективный по сравнению с классическими базами данных, использующими реляционный подход;
3. Выполнено оригинальное исследование моделей мышления в области обслуживания информационной структуры предприятия;

4. На основе разработанной в диссертации модели созданы архитектура системы и ее прототип;
5. Созданы специальные алгоритмы для анализа запросов пользователей и принятия решений;
6. Система, разработанная в рамках данной работы, включает в себя инновационные методы и алгоритмы поддержки принятия решений, использует модель мышления, базирующуюся на модели мышления Мински;
7. Представлена наглядная визуализация структуры области удаленной поддержки инфраструктуры.

Представленные в диссертации модель мышления, ее архитектура и реализация являются уникальными — на данный момент времени это единственная реализация модели мышления Мински.

Система, разработанная в диссертации, не является узкоспециализированной. Она также подходит для других областей, где требуется поддержка принятия решений. Например, при постановке медицинского диагноза, чтобы отбросить ложные диагнозы.

Кроме того, в систему можно загрузить данные о взаимосвязи органов человека и болезней. Далее, к каждому заболеванию добавить симптом и способ лечения, после этого можно делать запрос с симптомами, и система выдаст список вероятных заболеваний со способами их лечения.

В области диагностики проблем можно обучить систему узлам автомобиля, проблемам, с ними связанными, признаками этих проблем и способами их устранения.

## **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. — Т. 44, № 4. — С. 279–282.
2. *Toshchev A., Talanov M., Krehov A.* Thinking-Understanding approach in IT maintenance domain automation // *Global Journal on Technology: 3rd World*

*Conference on Information Technology (WCIT-2012)*. — 2013. — V. 3. — P. 879–894.

3. *Toshchev A., Talanov M.* Thinking model and machine understanding of English primitive texts and it's application in Infrastructure as Service domain // *Proceedings of Conference "Artificial Intelligence and Natural Language (AINL-2013)"*. — 2013. — С. 14–19. — <http://ainlconf.ru/material201303>.
4. *Toshchev A., Talanov M.* Архитектура и реализация интеллектуального агента для автоматической обработки входящих заявок с помощью искусственного интеллекта и семантических сетей // *Ученые записки Института социально-гуманитарных знаний*. — 2014. — Т. 2. — С. 288–292.
5. *Toshchev A., Talanov M.* Computational emotional thinking and virtual neurotransmitters // *International Journal of Synthetic Emotions (IJSE)*. — 2014. — V. 5. — P. 30–35.
6. *Toshchev A., Talanov M.* Appraisal, coping and high level emotions aspects of computational emotional thinking // *International Journal of Synthetic Emotions (IJSE)*. — 2015. — V. 06. — P. 65–72.
7. *Toshchev A.* Thinking model and machine understanding in automated user request processing // *CEUR Workshop Proceedings*. — 2014. — V. 1297. — P. 224–226.
8. *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. — V. 38. — P. 301–310.
9. *Тоцев А.С.* Возможности автоматизации разрешения инцидентов для области удаленной поддержки информационной инфраструктуры предприятия // *Экономика и менеджмент систем управления*. — 2015. — Т. 4. — С. 293–295.
10. *Тоцев А.С., Таланов М.О.* Вычислительная модель эмоций в интеллектуальных информационных системах // *Электронные библиотеки*. — 2015. — Т. 18. — С. 225–235.

11. *Тоцев А.С.* Применение моделей мышления в интеллектуальных вопросно-ответных системах // *Электронные библиотеки*. — 2015. — Т. 18. — С. 216–224.