

На правах рукописи



**Toshchev Alexander Sergeevich**

**INTELLECTUAL SYSTEM FOR INCREASING  
THE EFFICIENCY OF THE ENTERPRISE IT  
SERVICES**

Специальность 05.13.11 —  
«Математическое и программное обеспечение вычислительных  
машин, комплексов и компьютерных сетей»

**Автореферат**  
диссертации на соискание учёной степени  
candidate of Engineering Sciences

Kazan — 2017

Работа выполнена в Институте математики и механики (ИММ) им. Н.И. Лобачевского Казанского (Приволжского) федерального университета (КФУ)

Научный руководитель: доктор физико-математических наук, профессор, засл. деятель науки РТ, зав. кафедрой дифференциальных уравнений ИММ им. Н.И. Лобачевского КФУ  
**Елизаров Александр Михайлович**

Официальные оппоненты: **Райхлин Вадим Абрамович**,  
доктор физико-математических наук, профессор,  
Казанский национальный исследовательский технический университет им. А.Н. Туполева (КНИТУ-КАИ),  
профессор кафедры компьютерных систем

**Поляков Владимир Николаевич**,  
кандидат технических наук, доцент,  
Национальный исследовательский технологический университет МИСиС,  
доцент кафедры АСУ

Ведущая организация: Федеральный исследовательский центр «Информатика и управление» Российской академии наук (ФИЦ ИУ РАН), г. Москва

Защита состоится 25 мая 2017 года в 14:00 на заседании диссертационного совета Д 212.081.35 на базе ФГАОУ ВО Казанский (Приволжский) федеральный университет по адресу: 420008, Казань, ул. Кремлевская, 35.

С диссертацией можно ознакомиться в библиотеке ФГАОУ ВО «Казанский (Приволжский) федеральный университет» по адресу 420008, Казань, ул. Кремлевская, 35.

Автореферат разослан «    »                      2017 года.

Ученый секретарь

диссертационного совета

Д 212.081.35, канд. физ.-мат. наук, доцент

Еникеев Арслан Ильясович

## Общая характеристика работы

Область исследований, с которой связана диссертация, является комплексной и включает в себя различные направления работ, в частности, создания различных интеллектуальных систем. Сфера применения интеллектуальных систем обширна, например, в Институте Чиная (Индия) Е. Джубилсоном и П. Дханавантини ведутся исследования интеллектуальных систем обработки запросов пользователей в области телекоммуникаций, а в университете Ганновера (Германия) Р. Брунс и Дж. Данкель разрабатывают интеллектуальные системы для обработки запросов в службу спасения с целью уменьшения времени реакции на происшествие. В Санкт-Петербургском государственном университете (СПбГУ) под руководством В.И. Золотарева проводится оценка эффективности службы информационной поддержки в Вычислительном центре СПбГУ. В Сингапуре С. Фу и П. Леонг проведен анализ эффективности ИТ-службы поддержки крупной компании и показана возможность автоматизации ряда процессов (ИТ – информационные технологии).

Исследования в области интеллектуальных систем повышения эффективности ИТ-службы предприятия ведутся также лидерами ИТ-отрасли: компаниями HP <sup>1</sup> и IBM <sup>2</sup>. Например, известна многоцелевая интеллектуальная система IBM Watson, разработкой и исследованием которой занимается группа под руководством профессора А. Гоэля.

Для того чтобы система могла работать с запросами пользователя, она должна «понимать» язык, на котором они составлены. Подобные проблемы исследуются в области обработки естественного языка. Например, это подход GATE <sup>3</sup>, который активно развивается в университете Шеффилда (Великобритания) под руководством Г. Каллаган, Л. Моффат и С. Сзас. Другое направление — это семантический поиск, исследования в этой области также активно ведутся в университете Шеффилда, в частности, разработан подход Mimir, который реализует возможности поиска по принципу «поиск и открытие». Для организации поиска решений в соответствии с запросами пользователей в таких системах используются онтологии, например, широко применяется подход,

---

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

<sup>2</sup>Проект <http://www.ibm.com/smarterplanet/us/en/ibmwatson/>

<sup>3</sup>Проект <https://gate.ac.uk/>

предложенный С. Дей и А. Джеймс из Калифорнийского университета (США), основанный на применении деревьев тегов в онтологии.

Для придания интеллектуальной системе гибкости необходимо дать ей возможность проводить логические рассуждения. Одной из ведущих организаций в этом направлении исследований является консорциум OpenCog <sup>4</sup> (США). Этими работами руководит Бен Герцель (председатель Artificial General Intelligence Society и OpenCog Foundation) — один из мировых лидеров в области искусственного интеллекта. Исследования в области машинной логики также ведутся в рамках проекта NARS <sup>5</sup> под руководством профессора университета Темпла (США) Пея Вонга.

**Целью** диссертации является разработка интеллектуальной системы повышения эффективности деятельности ИТ-службы предприятия.

**Область исследования** — разработка систем управления базами данных и знаний.

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

**Методы исследования** — теоретические методы: имитационное моделирование, теория баз знаний в области искусственного интеллекта; специальные методы: системное моделирование; экспериментальные методы: метод наблюдений, проведение экспериментов.

Для достижения поставленной цели были решены следующие проблемы и задачи:

1. Провести анализ систем управления базами знаний в области поддержки информационной инфраструктуры предприятия;
2. Разработать и построить модель проблемно-ориентированной системы управления базой знаний для принятия решений и оптимизации процессов регистрации, анализа и обработки запросов пользователей в области обслуживания информационной инфраструктуры предприятия;
3. На основе построенной модели разработать архитектуру и создать прототип интеллектуальной системы повышения эффективности деятельности ИТ-службы предприятия;
4. Провести апробацию прототипа на тестовых данных.

---

<sup>4</sup>Проект <http://opencog.org/>

<sup>5</sup>Проект <https://sites.google.com/site/narswang/>

### **Основные положения, выносимые на защиту:**

1. Результаты анализа систем управления базами знаний в области поддержки ИТ-инфраструктуры предприятия;
2. Построенная модель проблемно-ориентированной системы управления базой знаний и оптимизации процессов обработки запросов пользователей в области обслуживания ИТ-инфраструктуры предприятия;
3. Созданный прототип программной реализации модели проблемно-ориентированной системы управления базой знаний и оптимизации обработки запросов пользователей в области обслуживания ИТ-инфраструктуры предприятия;
4. Результаты апробации прототипа проблемно-ориентированной системы управления на контрольных примерах.

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

1. На основе обобщения модели мышления, разработанной М. Мински, создана имитационная модель проблемно-ориентированной системы управления, принятия решений в области обслуживания ИТ-инфраструктуры предприятия;
2. Исследованы возможности использования моделей мышления применительно к области обслуживания информационной инфраструктуры предприятия;
3. Представлены новая схема данных и оригинальный способ хранения данных для построенной модели мышления, более эффективный по сравнению со стандартными способами хранения (такими, например, как реляционные базы данных);
4. На основе построенного обобщения модели мышления Мински созданы архитектура системы обслуживания информационной инфраструктуры предприятия и программный прототип этой системы.

**Практическая значимость.** Система, разработанная в рамках диссертации, имеет значимый практический характер. Идея работы зародилась под влиянием производственных проблем в ИТ-отрасли, с которыми автор сталкивался ежедневно в процессе разрешения различных инцидентов, возникающих в деятельности службы технической поддержки ОАО «АйСиЭл КПО-ВС (г. Казань)» — одним из крупнейших системообразующих предприятий ИТ-отрасли

Республики Татарстан. Поэтому было необходимо выработать глубокое понимание конкретной предметной области, чтобы выбрать приемлемое программное решение, получившее практическое применение при организации 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, 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 05.13.11 — Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей, сопоставление приведено в таблице 1.

Таблица 1 — Сопоставление направлений исследований предусмотренных специальностью 05.13.11, и результатов, полученных в диссертации

Направление исследования	Результат работы
Языки программирования и системы программирования, семантика программ	Разработана семантическая модель организации хранения знаний
Системы управления базами данных и знаний	Разработан прототип Thinking Understanding (TU) системы хранения знаний и принятия решений в сфере поддержки ИТ-инфраструктуры предприятия, который был испытан на модельных данных
Модели и методы создания программ и программных систем для параллельной и распределенной обработки данных, языки и инструментальные средства параллельного программирования	Разработан метод параллельной обработки экспертной информации с возможностью обучения при помощи прототипа TU

**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)». Ожидаемым был результат в 51% (доля разрешенных проблем, поставленных пользователями), но и достигнутый результат в 30% мы считаем приемлемым, так как он значительно увеличивает эффективность разрешения проблемных запросов.

**Личный вклад.** Автор провел анализ запросов пользователей и классифицировал их; построил модель целевой области и выявил возможности оптимизации процессов в ней. Данные для исследования (выгрузка из систем регистрации запросов пользователей ICL) были получены при помощи А.В. Крехова. Совместно с М.О. Талановым автор 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 10 печатных изданиях [?; ?; ?; ?; ?; ?; ?; ?; ?; ?], из которых статьи [?; ?] проиндексированы в БД Scopus и входят в перечень журналов ВАК РФ, статья [?] проин-

дексирована также в БД Web of Science, работа [?] опубликована в журнале из перечня ВАК РФ, статья [?] проиндексирована в БД РИНЦ, работы [?; ?; ?] опубликованы в материалах международных и всероссийских конференций, статьи [?; ?] опубликованы в международном журнале "International Journal of Synthetic Emotions", входящем в индекс ACM (Association for Computing Machinery).

В работе [?] А.С. Тощев предложил оригинальную идею автоматического конструирования приложений. В статье [?] А.С. Тощевым был разработан программный комплекс, М.О. Таланов предложил идею, а А.В. Крехов предоставил тестовые данные из системы регистрации запросов службы технической поддержки ИТ-инфраструктуры ОАО «АйСиЭл КПО-ВС (г. Казань)». В работе [?] А.С. Тощев предложил и реализовал архитектуру интеллектуального агента, М.О. Таланов поставил задачу проверки результатов реализации этого подхода. В статьях [?; ?] А.С. Тощев выполнил проверку модели, предложенной М.О. Талановым. В работе [?] А.С. Тощев реализовал модель. В статье [?] А.С. Тощев выполнил доработку модели мышления, М.О. Таланов поставил задачу придания универсальности системе. В статье [?] А.С. Тощев проанализировал результаты работы системы регистрации запросов службы технической поддержки ИТ-инфраструктуры ОАО «АйСиЭл КПО-ВС (г. Казань)» и выдвинул гипотезу о возможности автоматизации разрешения части запросов. В работах [?; ?] А.С. Тощев провел разработку и проверку модели, М.О. Таланов разработал основную концептуальную идею.

## Содержание работы

Во **введении** обоснована актуальность исследований, проведенных в рамках диссертации; даны общая характеристика работы и анализ исследований в области обслуживания информационной инфраструктуры предприятия; проведен обзор и на основе выявленного роста публикационной активности в рассматриваемой предметной области (по данным Scopus) обоснована актуальность проведенных исследований.

**Первая глава** диссертации посвящена постановке задачи и обзору интеллектуальных систем регистрации и анализа проблемных ситуаций, возникающих в ИТ-инфраструктуре предприятия. Глава начинается с описания модели теории массового обслуживания (ТМО) для служб, занимающихся устранением проблемных ситуаций, возникающих в ИТ-инфраструктуре предприятия. На



рисунке 1 представлена модель системы массового обслуживания, в которой использованы следующие обозначения:  $\lambda$  — интенсивность входящего потока;  $\alpha$  — доля заявок, для которых время ожидания в очереди превышает  $\max(T_q)$ ;  $\mu$  — величина, обратная среднему времени нахождения заявки у агента;  $n$  — число агентов;  $T_q$  — время нахождения заявки в очереди в часах;  $SLA$  — уровень обслуживания или доля заявок, для которых время в очереди не превышает  $\max(T_q)$ ,  $SLA = 1 - \alpha$ ;  $T_p$  — время удовлетворения заявки;  $\alpha_n$  — количество заявок;  $T_{qp} = T_q + T_p$  — время прохождения заявки через систему;  $S(\mu) = \frac{R_p}{\mu}$  — средняя стоимость выполнения одной заявки;  $R_p$  — средняя стоимость часа работы специалиста (выводится далее).

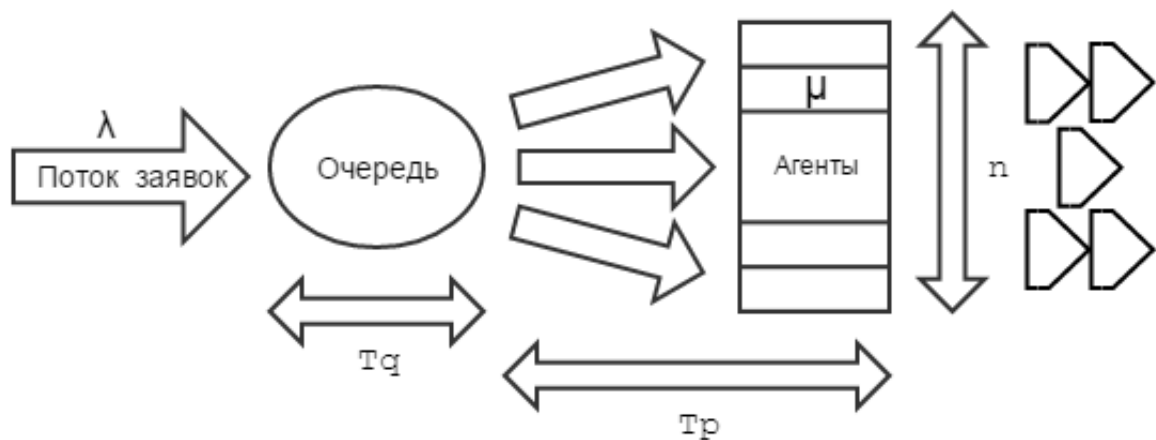


Рис. 1 — Модель системы массового обслуживания.

Существует несколько подходов решения задач ТМО:

- Аналитическое решение для простейших систем, которое позволяет выразить  $T_q(t)$  через  $\lambda$ ,  $\mu$  и  $n$ ;
- Решение с помощью имитационного подхода, где строится гистограмма  $T_q(t)$ , по которой оценивается достаточность  $n$  для обеспечения  $SLA$ ;
- Решение с помощью эконометрического подхода, которое подходит для систем с достаточно большим  $n$ . В таких системах возможно оценить  $T_q(t)$  по имеющейся статистике.

На основе комбинации формул Эрланга, модели Энгсета и модели Полячека – Хинчина построена формула для решения задач ТМО на основе аналитического подхода путем нахождения распределения вероятностей для  $T_{qp}$ . Основной же задачей этой работы является прогнозирование необходимых ресурсов для максимизации  $SLA$  ( $SLA = 1 - \alpha$ ). В главе 1 диссертации рассмотрена

задача минимизации  $T_{qp}$ ,  $S(\mu)$  и динамического выделения ресурсов. На основе статистики, собранной в компании ОАО «АйСиЭл КПО-ВС (г. Казань)», был подсчитан следующий коэффициент  $T_{qp} = 47,9$  при  $n = 6$ ;  $SLA = 0,82$ ;  $\alpha = 0,18$ ;  $\alpha_n = 2920$ .

В главе 1 также 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 в данной главе представлен сравнительный анализ методов и программных комплексов обработки текстов.

При проведении анализа были использованы следующие средства обработки естественного языка: Open NLP<sup>6</sup>, Relex<sup>7</sup>, StanfordParser<sup>8</sup>. Оценка качества функционирования этих средств проводилась при помощи метрик, представленных в таблице 2, а полученные результаты приведены на рисунке 2. Как видно, наилучший результат по всем трем метрикам показала система Relex, она и была выбрана в качестве основного средства обработки естественного языка.

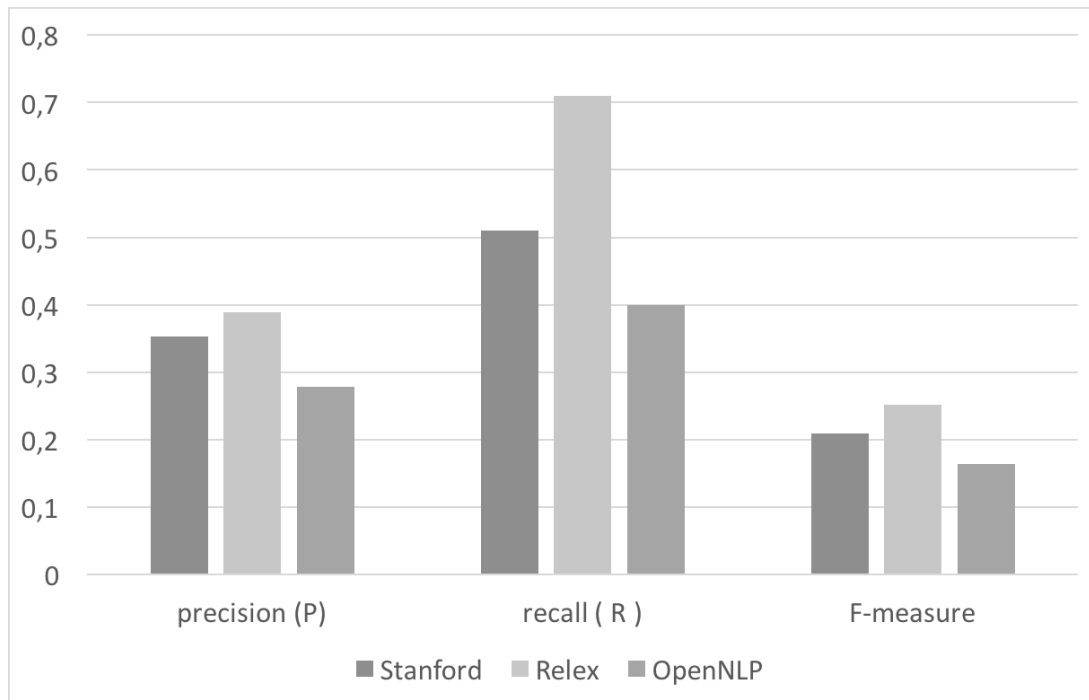


Рис. 2 — Результаты анализа средств обработки естественного языка

<sup>6</sup>Проект <http://opennlp.apache.org>

<sup>7</sup>Проект <http://opencog.org>

<sup>8</sup>Проект <http://nlp.stanford.edu/software/lex-parser.shtml>

Таблица 2 — Таблица метрик

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

Кроме того, в главе 1 проведен анализ существующих программных систем автоматизации в области поддержки информационной инфраструктуры предприятия: HP Open View<sup>9</sup>, ServiceNOW<sup>10</sup>, IBM Watson<sup>11</sup>. Установлено, что все рассмотренные системы не соответствуют полному набору необходимых требований, приведенных во введении. Таблица 3 содержит сводные данные по рассмотренным системам — указаны наличие или отсутствие у них той или иной функции. 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

<sup>9</sup>[https://ru.wikipedia.org/wiki/HP\\_OpenView](https://ru.wikipedia.org/wiki/HP_OpenView)

<sup>10</sup><http://www.servicenow.com/>

<sup>11</sup><http://www.ibm.com/smarterplanet/us/en/ibmwatson/>

**Таблица 3 – продолжение**

<b>Сравнительный пункт</b>	<b>HP Open View</b>	<b>ServiceNOW</b>	<b>IBM Watson</b>
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:

- 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.

Отметим, 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 описана подробно.

**Модель Menta 0.1, построенная с использованием деревьев принятия решений**, была одной из первых, которая была опробована. При построении модели использованы 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 характеризуется 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 экспериментов было выявлено отсутствие устойчивости к ошибкам входной информации: грамматическим и содержательным. Например, входной файл не имел отношения к программной системе, модель которой содержалась в базе знаний; 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 решения. Система Menta 0.3 содержала, как составляющую часть, модель целевого приложения (как и Menta 0.1) и список решений тех или иных проблем (How-To), найденных ранее. При помощи генетического алгоритма модель строила решение, проверяла его при помощи логического движка NARS<sup>12</sup> на соответствие критериям, заданным пользователем. С точки зрения генетических алгоритмов, такая проверка — функция отбора особей из поколения.

В результате проведенных экспериментов были выявлены следующие проблемы: отсутствие обучения; отсутствие обработки естественного языка; после апробации оказалось, что список критериев соответствия решения требованиям пользователя (набор правил) 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>13</sup>, сохранила следующие основные концептуальные элементы предыдущих моделей и

---

<sup>12</sup><http://www.cogsci.indiana.edu/farg/peiwang/papers.html>

<sup>13</sup>[https://en.wikipedia.org/wiki/The\\_Emotion\\_Machine](https://en.wikipedia.org/wiki/The_Emotion_Machine)

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 по функциональности компоненты предыдущих систем. Реализованная модель названа TU (от англ. "Thinking Understanding" — «мышление и понимание»).

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

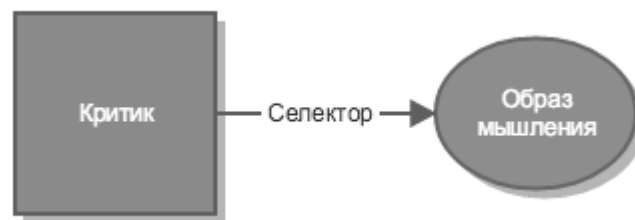


Рис. 3 —  $T^3$

*Критик* представляет собой определенный переключатель, который срабатывает при определенных событиях. 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 множества критиков: выполнить расчет, уточнить технические детали. Кроме того, параллельно может активироваться критик контроля уровня загруженности, сообщающий о необходимости отдыха.

*Селектор* занимается выбором необходимых ресурсов, которыми, например, могут быть: критик, образ мышления.

*Образ мышления* — это способ решения проблемы. Он может быть сложным и, например, может активировать других критиков. Так, размышляя над проблемой, специалист понимает, что нужно рассмотреть все возможные комбинации, и тут он решает поискать готовое решение: может быть кто-то уже рассмотрел все возможные комбинации, и можно будет его использовать. Здесь «поиск готового решения» является критиком внутри образа мышления «поиск решения».

На рисунке 4 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, появилась проблема, связанная с отсутствием доступа, значит, нужно запустить служебную программу для предоставления прав пользователю. Под ресурсами здесь понимается набор знаний из базы знаний: критики, селекторы, образы мышления, готовые решения.

Если активировалось много критиков, то проблему нужно уточнить, иначе степень неопределенности будет слишком высокой. Если проблема очень похожа на уже проанализированную, то можно действовать и судить по аналогии.

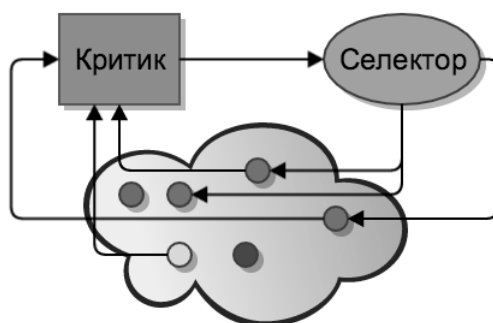


Рис. 4 —  $T^3$  в разрезе ресурсов

Другой важной частью 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, основные из которых состоят в следующем.

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»

**Таблица 4 – продолжение**

<b>Уровень</b>	<b>Описание</b>
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?»

Для программной экспертной системы очень важно обладать способностью мыслить и рассуждать, например, действовать по аналогии. Множество запросов типично, запросы отличаются лишь параметрами. Например, таковым является запрос «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 соответствует целевой области поддержки ИТ-инфраструктуры предприятия. На основе подхода Мински построена модель системы, которая реализует основные функции: обучение, понимание инцидента, поиск решения, применение решения.

В третьей главе описаны архитектура и реализация 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 компонентов Critic (Критик), привязанный к данной цели (HelpUser);
  4. Активируется компонент PreliminaryAnnotator (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 текущей 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 Tehcnical 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, информирующий об этом.
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, появившийся 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 оценки эффективности работы модели, полученные на основе проведенных экспериментов. Были проведены тесты для выполнения сравнения с работой специалиста-человека. Был выбран контрольный список запросов пользователя (инцидентов). Сравнивалась скорость разрешения инцидента. 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
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

**Таблица 6 – продолжение**

<b>Инцидент</b>	<b>TSS1 (.мс)</b>	<b>TU (.мс)</b>
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 — ком-муникатор Cisco IP. Пожалуйста, помогите исправить проблему с ИП-телефоном, он застревает во время конфигурирования и иногда показывает ошибку «Безопасность»)	13000	12343

Показатели, приведенные во введении, приобрели следующие значения  $T_{qp}=32,9$  при  $n=8$ ;  $SLA=0,96$ ;  $\alpha=0,04$ ;  $\alpha_n=2920$ .

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

Решены следующие задачи и достигнуты следующие результаты.

1. Создана модель проблемно-ориентированной системы управления знаниями в области обслуживания информационной инфраструктуры предприятия на основе обобщения модели мышления;
2. Представлены новая модель данных для модели мышления и оригинальный способ их хранения, более эффективный по сравнению с классическими базами данных, использующими реляционный подход;
3. Выполнено оригинальное исследование моделей мышления в области обслуживания информационной инфраструктуры предприятия;
4. На основе модели, разработанной в диссертации, созданы архитектура системы и ее прототип;
5. Система, разработанная в рамках данной работы, включает в себя инновационные методы и алгоритмы поддержки принятия решений, использует обобщенную модель мышления Мински;
6. Представлена 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 и подходит для других областей, где требуется организация базы знаний, например, при постановке медицинского диагноза, чтобы отбросить ложные диагнозы.

В области диагностики проблем можно обучить систему сведениям об узлах автомобиля и проблемах, с ними связанных, признаках этих проблем и способах их устранения.

Работа выполнена частично за счет средств субсидии, выделенной Казанскому федеральному университету для выполнения государственного задания в сфере научной деятельности, проекты 1.2368.2017, «Бюджет 17-97».