

Казанский (Приволжский) Федеральный Университет

На правах рукописи

УДК **xxx.xxx**

Тощев Александр Сергеевич

**Разработка эффективного подхода обработки производственных задач
прикладного характера в области обслуживания программного
обеспечения и информационной инфраструктуры предприятия на основе
стохастического поиска, вероятностно-логических рассуждений и
машинного обучения**

Специальность 05.13.01 —

«Системный анализ, управление и обработка информации (по отраслям)»

Диссертация на соискание учёной степени

Кандидат технических наук

Научный руководитель:

уч. степень, уч. звание

Елизаров А.М.

Казань — 2015

Оглавление

Введение	4
1 ПОСТАНОВКА ЗАДАЧИ ПОЛУЧЕНИЯ, АНАЛИЗА И ОБРАБОТКИ ЭКСПЕРТ-НОЙ ИНФОРМАЦИИ	6
1.1 Возникновение области	6
1.2 Прогноз развития области	7
1.3 Методологии, используемые в области IT аутсорсинга: ITIL и ITSM	8
1.4 Постановка задачи	8
2 МЕТОДЫ И КОМПЛЕКСЫ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА	10
2.1 Обработка Эталонных Текстов	10
2.2 Обработка текстов с ошибками	12
2.3 Сравнение средств обработки русского и английского языка	13
2.4 Вывод по главе	15
3 АНАЛИЗ ТЕКУЩИХ РЕШЕНИЙ ПОЛУЧЕНИЯ, АНАЛИЗА И ОБРАБОТКИ ЭКСПЕРТНОЙ ИНФОРМАЦИИ В ОБЛАСТИ ОБСЛУЖИВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ	16
3.1 Обзор решений	16
3.2 Требования к системе	17
4 ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ СИСТЕМЫ, АРХИТЕКТУРА И РЕАЛИЗАЦИЯ	19
4.1 Модель мышления Марвина Мински	19
4.1.1 Крити-Селектор-Путь мышления	19
4.1.2 Уровни мышления	20
Заключение	23
Список литературы	24
Список рисунков	26
Список таблиц	27
А Название первого приложения	28

В Очень длинное название второго приложения, в котором продемонстрирована ра-	
бота с длинными таблицами	29
В.1 Подраздел приложения	29
В.2 Ещё один подраздел приложения	31
В.3 Очередной подраздел приложения	32
В.4 И ещё один подраздел приложения	32

Введение

В настоящее время в области IT набрало 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 потребность автоматизации работы. В данном контексте рассматривается автоматизация обработки инцидентов, начиная с разбора инцидентов на естественном языке и заканчивая поиском решения и применением решения. Главными требованиями к системе являются

1. Обработка естественного языка
2. Возможность обучения
3. Общение с специалистом
4. Проведение логических рассуждений: аналогия, дедукция, индукция
5. Умения абстрагировать решение и экстраполировать его на другие решения

На данный момент многие компании ведут разработку подобных систем. Примером такой системы является набирающая популярность система IBM Watson [1]. Подобный класс системы также называют вопросно-ответными системами. Другим примером является система Wolfram Alpha [2]. В данной работе был сделан акцент на попытку создания мыслящей системы на основе модели мышления Марвина Мински [3].

Целью данной работы является создание архитектуры и реализация базового прототипа программного комплекса обеспечивающего разбор и формализацию входного запроса пользователя и поиск решения данной проблемы.

Для достижения поставленной цели необходимо было решить следующие задачи:

1. Исследовать целевую область
2. Вычислить возможность автоматизации целевой области
3. Исследовать модель мышления Марвина Мински
4. Адаптировать модель для прикладной реализации
5. Создать архитектуру приложения на основе модели

6. Реализовать прототип на основе архитектуры

Основные положения, выносимые на защиту:

1. Возможность автоматизации области предоставления удаленной поддержки информационной инфраструктуры
2. Прикладное применение модели мышления Марвина Мински для решения задачи автоматизации
3. Возможность программной реализации модели мышления Марвина Мински
4. Экстраполяция программной системы для других областей

Научная новизна:

1. Впервые была представлена реализация модели мышления Мински на практике
2. Была представлена новая модель данных для модели мышления
3. Было выполнено оригинальное исследование модели мышления ...

Научная и практическая значимость ...

Степень достоверности полученных результатов обеспечивается результатами выполнения тестов на контрольных примерах. Результаты находятся в соответствии с результатами, полученными другими авторами и экспертными системами

Апробация работы. Основные результаты работы докладывались на:

- RCDL-2014
- AINL-2013
- WCIT-2012

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

Публикации. Основные результаты по теме диссертации изложены в XX печатных изданиях [4–8], X из которых изданы в журналах, рекомендованных ВАК [4–6], XX — в тезисах докладов [7, 8].

Объем и структура работы. Диссертация состоит из введения, четырех глав, заключения и двух приложений. Полный объем диссертации составляет XXX страница с XX рисунками и XX таблицами. Список литературы содержит XXX наименований.

Глава 1

ПОСТАНОВКА ЗАДАЧИ ПОЛУЧЕНИЯ, АНАЛИЗА И ОБРАБОТКИ ЭКСПЕРТНОЙ ИНФОРМАЦИИ

1.1. Возникновение области

В настоящее время в области IT набрало 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. Большинство проблем, которые решает удаленная служба поддержки носят весьма тривиальный характер :

- Установить приложение
- Переустановить приложение
- Решить проблему с доступом к тому или иному ресурсу

Данные проблемы решают специалисты технической поддержки. Обычно техническая поддержка делится на несколько линий:

1. Первая линия. Решение уже известных, задокументированных проблем, работа напрямую с пользователем
2. Вторая линия. Решение ранее неизвестных проблем
3. Третья линия. Решение сложных и нетривиальных проблем
4. Четвертая линия. Решение архитектурных проблем инфраструктуры

Каждая линия поддержки представлена специалистами. В среднем команда, обслуживающая одного заказчика насчитывает 60 человек. Процентное соотношение специалистов разных линий поддержки отображено на Диаграмме 1.1

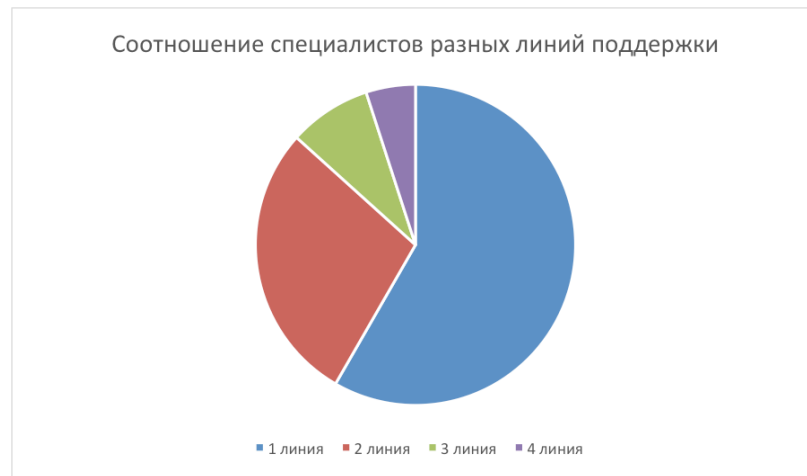


Рисунок 1.1: Диаграмма состава команд

Работа специалиста 1 линии поддержки состоит из множества рутинных и простых задач. На Диаграмме 1.2 показано соотношение разных типов проблем, встречающихся во время работы поддержки

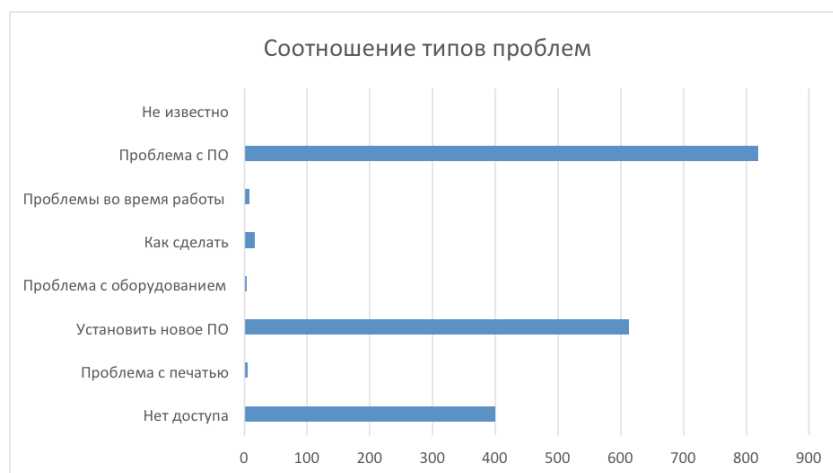


Рисунок 1.2: Диаграмма соотношений типов проблем

Решение части задач может быть автоматизировано, а специалисты получают дополнительное время на решение более интересных задач. Проблема заключается в автоматизации решения рутинных задач в области удаленной поддержки инфраструктуры.

1.2. Прогноз развития области

Основной тенденцией в развитии области удаленной поддержки инфраструктуры является попытки удешевить и улучшить стоимость предоставления услуг.

Компании, работающие на этом рынке вкладывают большие деньги в автоматизацию. Кроме того современное развитие науки и техники, а точнее вычислительных мощностей позволяет автоматизацию даже самых наукоемких процессов.

Дальнейшим развитием области является замена человеческих специалистов на автоматические

Таблица 1.1: Категории инцидентов

Категория	Описание
Проблема с ПО	Проблема при запуске ПО на компьютере. Решается переустановкой
Проблемы во время работы	Проблема с функционированием программного обеспечения
Как сделать	Запрос на инструкцию по работе с тем или иным компонентом рабочей станции
Проблема с оборудованием	Неполадки на уровне оборудования
Установить новое ПО	Требование установки нового программного обеспечения
Проблема с печатью	Установка принтера в систему
Нет доступа	Нет доступа к общим ресурсам

системы. Многие ведущие компании ведут разработки в этом направлении. Например, компания HP. Данная компания имеет свою системы по регистрации подобных инцидентов и сейчас ведется работа над автоматизацией системы.

1.3. Методологии, используемые в области IT аутсорсинга: ITIL и ITSM

В области IT аутсорсинга есть несколько готовых стандартов ведения работ. Одним из таких стандартов является библиотека ITIL. Данный стандарт описывает лучшие практики организации работ в области IT аутсорсинга. Используемый в библиотеки подход соответствует стандартам ISO 9000 (ГОСТ Р ИСО 9000). Наличие стандартов в области диктует унифицированность постановки проблем, а также унифицированность алгоритмов решения. Такие предпосылки говорят о возможности частично или в некоторых случаях полной автоматизации решения проблем.

1.4. Постановка задачи

Задачами данного исследования являются:

- Изучение возможности автоматизации области удаленной поддержки инфраструктуры путем анализа области
- Выработка критериев и сравнительный анализ существующих решений в области
- Создание программного комплекса (фреймворка) для автоматизации поддержки удаленной инфраструктуры

- Подсчет статистических результатов работы комплекса

Глава 2

МЕТОДЫ И КОМПЛЕКСЫ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА

2.1. Обработка Эталонных Текстов

В данном разделе проводится обзор обработчиков естественного языка. За основу были взяты инциденты из выгрузки систем поддержки ОАО "ICL КПО-ВС".

Ввиду специфики области основным языком был выбран английский язык. Был сформирован список из типичных эталонных фраз, на которых тестировались обработчики естественного языка. Фразы были выявлены путем анализа существующих отчетов об инцидентах. Примерами инцидентов являются следующие инциденты:

Инцидент 1 *User had received wrong application. User has ordered Wordfinder Business Economical for her service tag 7Q4TC3J, there is completed order in LOT with number ITCOORD-18125. However she received wrong version, she received Wordfinder Tehcnical instead of Business Economical. Please assist.*

Инцидент 2 *Laptop – user has almost full C: but when he looks in the properties of the files and folders on C: they are only 40GB and he has a 55GB drive.*

Инцидент 3 *User cannot find Produkt Manageron start menu. Please reinstall.*

Инцидент 4 *User needs to have pdf 995 re-installed please.*

Во время анализа были использованы следующие обработчики естественного языка:

1. Open NLP [9]
2. ReLEx [10]
3. StanfordParser [11]

Результат работы вычислялся при помощи метрик, представленных в Таблице 2.1.

Результаты приведены на сводной диаграмме Рисунок 2.1

Таблица 2.1: 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	$T = Ac + R$

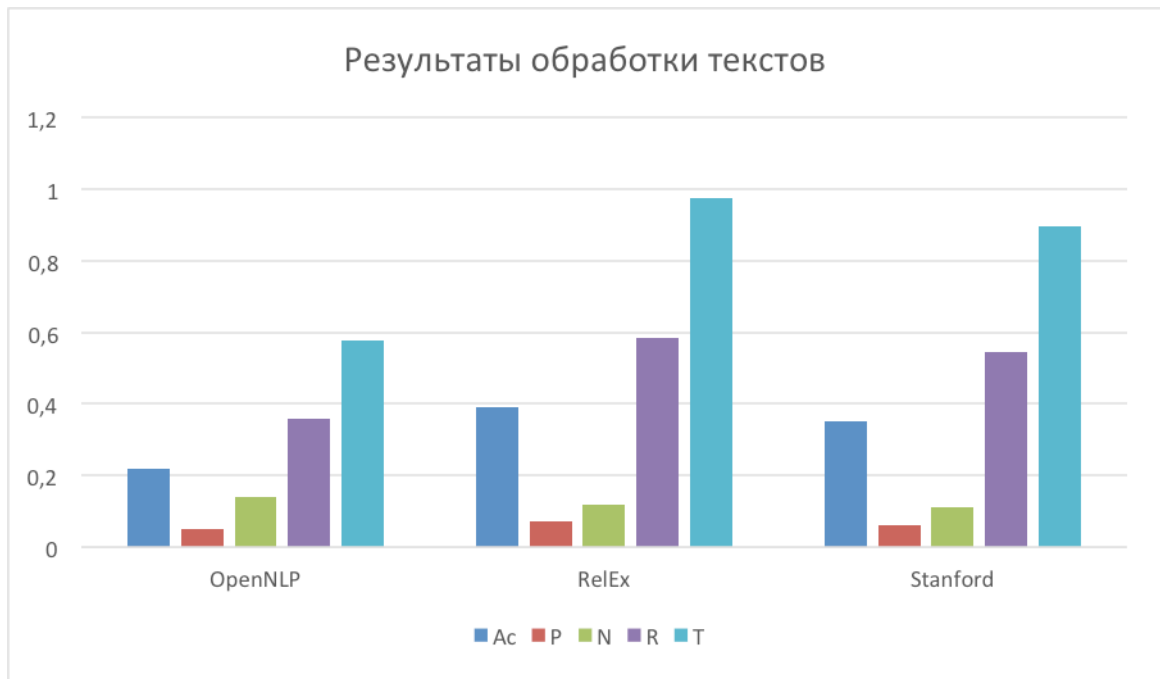


Рисунок 2.1: Результаты обработки текстов

Из диаграммы видно, что наилучшие результаты показывает обработчик RelEx [10]. После анализа необработанных инцидентов было выявлено несколько проблем у всех обработчиков:

1. Невозможности корректировки простых грамматических ошибок, связанных с пропущенными пробелами или неверным форматированием. Ошибки первого типа.
2. Ошибки неверной интерпретации слов в предложении. Например, слово *please* интерпретировалось как глагол, хотя является по смыслу «формой вежливости». Ошибки второго типа.

2.2. Обработка текстов с ошибками

По результатам прошлого раздела было решено выбрать в качестве обработчика естественного языка RelEx, но были выявлены некоторые проблемы. Было принято решение исправить данные проблемы при помощи предварительной обработки текста. Предварительная обработка текста была разбита на несколько фаз:

1. Комплексная корректировка ошибок
2. Обработка при помощи внутренней базы знаний

Для того, чтобы избавиться от орфографических, синтаксических ошибок используется составной корректировщик. Данный компонент имеет модульную структуру и применяет корректировку последовательно.

Для данного компонента были написаны модули корректировки:

- Google API

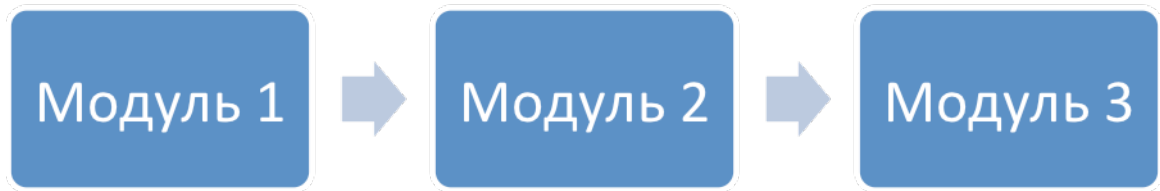


Рисунок 2.2: Архитектура предварительной обработки текста

– After The Deadline

Таким образом удалось исправить большинство ошибок, связанных с синтаксисом, грамматикой, орфографией. Также удалось исправить ошибки неверного написания: лишних пробелов, пропущенных запятых, пропущенных точек. По-прежнему остается проблема обработки неверной интерпретации слов в тексте.

Для корректировки ошибок второго типа было использовано инъекция в работу обработчика RelEx. Ввиду OpenSource природы проекта, модульности был подменен модуль извлечения и обработки слов в предложении. Стандартный процесс обработки был разбит на «предобработку» и «обработку». Стадия «обработки» включала в себя алгоритм работы такой же как был до этого в модули, на стадии «предобработки» управление передается модулю основного приложения, который проверяет данное слово на предмет его вхождения во внутреннюю Базу Знаний и если таковое имеется, то приложение передает соответствующие корректировки в модуль

2.3. Сравнение средств обработки русского и английского языка

Средства обработки естественного языка принято относить к большому классу средств NLP – Natural Language Processing. Для английского языка существует множество открытых средств обработки естественного языка, для русского языка найти их гораздо сложнее. Рассмотрим архитектуру средств обработки естественного языка на примере OpenCog RelEx.

OpenCog RelEx использует результаты обработки Link Grammar [12]. Link Grammar поддерживает

множество языков: английский, русский, турецкий, немецкий и т.д. RelEx использует вывод LG и преобразует его в формат связей. **Пример 1.** User is unable to start KDP web, please reinstall Java.

Результат

```

_obj(start, KBP)
pos(start, verb)
inflection-TAG(start, .v)
tense(start, present)
pos([web], WORD)
noun_number(KBP, singular)
definite-FLAG(KBP, T)
pos(KBP, noun)
_advmod(reinstall, please)
pos(reinstall, verb)
inflection-TAG(reinstall, .v)
tense(reinstall, present)
pos(please, adv)
inflection-TAG(please, .e)
noun_number(Java, singular)
definite-FLAG(Java, T)
pos(Java, noun)
pos(., punctuation)
_obj(,, Java)
pos(,, verb)
tense(,, infinitive)
HYP(,, T)
_to-do(unable, ,)
pos(unable, adj)
inflection-TAG(unable, .a)
tense(unable, present)
pos(to, prep)
inflection-TAG(to, .r)
pos(be, verb)
inflection-TAG(be, .v)
_predadj(User, unable)
noun_number(User, singular)
definite-FLAG(User, T)
pos(User, noun)

```

Возьмем разбор слова `start`. В результате мы получаем несколько отношений:

- `pos(start, verb)` - `start` глагол
- `tense(start, present)` - время настоящее
- `inflection-TAG(start, .v)` - метод обозначения на схеме (индекс)

На их основе можно формализовать приложение на естественном языке. Остальные парсеры пока не поддерживают русский язык. Существуют открытые проекты, но они еще недостаточно развиты.

2.4. Вывод по главе

В данной главе был выполнен основной анализ обработчиков естественного языка. Ввиду развитости и доступности было решено использовать OpenCog RelEx.

Глава 3

АНАЛИЗ ТЕКУЩИХ РЕШЕНИЙ ПОЛУЧЕНИЯ, АНАЛИЗА И ОБРАБОТКИ ЭКСПЕРТНОЙ ИНФОРМАЦИИ В ОБЛАСТИ ОБСЛУЖИВАНИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ

3.1. Обзор решений

HP OpenView [13] является комплексным программным решением по мониторингу ИТ инфраструктуры предприятия. Система имеет множество модулей. Данная система охватывает широкий спектр возможностей:

- Мониторинг
- Регистрация инцидентов
- Управление системами

Система не поддерживает:

- Понимание и формализация запросов
- Автоматическое исправление проблемы на основе формализации запроса

ServiceNOW Средства автоматизации сервиса. Предоставляет следующие возможности:

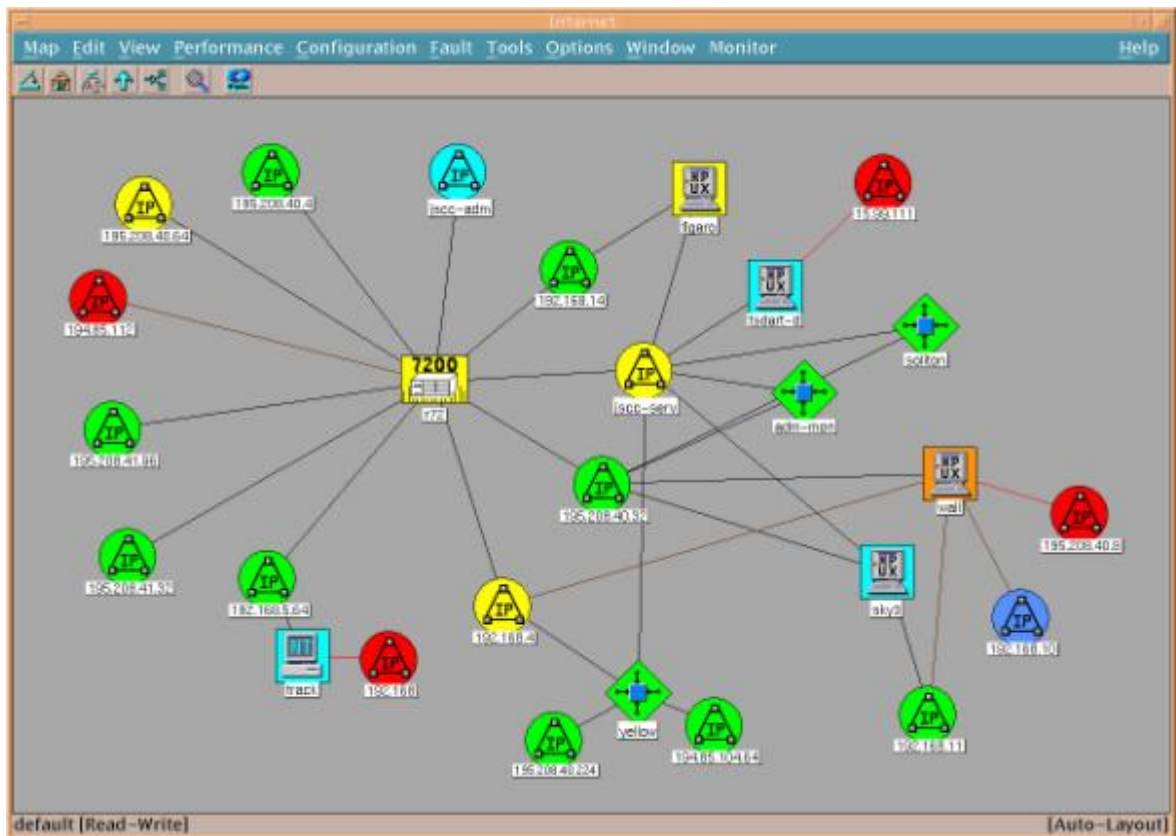


Рисунок 3.1: HP OpenView

- Регистрация инцидентов
- Создание цепи обработки инцидента

Система не поддерживает:

- Понимание и формализация запросов
- Автоматическое исправление проблемы на основе формализации запроса

Прочие системы Кроме того существуют дополнительные способы автоматизации

- Обработка инцидентов посредством регулярных выражений. В таком решении нет гибкости, так как обработка идет путем поиска ключевых слов вне контекста
- Обработка инцидентов при помощи скриптов. Автоматизирует лишь рутинные операции

3.2. Требования к системе

Основными требованиями к системе являются следующие:

- Понимание входящей информации на естественном языке
- Формализация инцидента в контексте

Number	Short description	Approval	Type	State	Planned start date	Planned end date
CHG00000001	Rollback Oracle Version	Approved	Comprehensive	Open	2014-12-23 15:00:00	2014-12-23 17:00:00
CHG00000002	Switch Sales over to the new 555 prefix	Not Yet Requested		Work in Progress	2014-12-24 11:00:00	2014-12-24 11:30:00
CHG00000003	Roll back Windows SP2 patch	Approved		Work in Progress	2014-12-29 17:00:00	2014-12-29 22:00:00
CHG00000004	Upgrade to Oracle 11i	Not Yet Requested		Open	2014-12-23 23:00:00	2014-12-24 05:00:00
CHG00000005	Install new PBX	Approved		Open	2014-12-26 23:00:00	2014-12-30 15:46:31
CHG00000006	Put another 100 Gb drive on the 2nd Floor Server	Not Yet Requested		Open	2014-12-31 22:00:00	2014-12-31 22:45:00
CHG00000007	R&D wants to know what it'd cost to switch them over to Linux desktops	Rejected		Open	2015-01-01 15:00:00	2015-01-01 19:00:00
CHG00000008	Install new Cisco	Requested		Open	2014-12-29 11:30:00	2014-12-29 18:30:00
CHG00000009	Apply patches 10.2.0.1 to 10.2.0.3	Approved	Comprehensive	Open	2014-12-27 02:00:00	2014-12-27 07:00:15
CHG00000010	Java Application Server change	Approved	Comprehensive	Open	2015-01-16 02:00:00	2015-01-16 07:00:00

Рисунок 3.2: Service NOW

- Поиск решение инцидента
- Обучение решению инцидента
- Умение проводить логические рассуждения: генерализацию, специализацию, синонимичный поиск
- Умение мыслить

Требования к системе формировались исходя из возможностей специалистов поддержки, а также анализа проблем, которыми они занимаются. Большинство инцидентов тривиальные и типичные, но все они разные. Для человека проблема "Please insall Firefox" и "Please install Chrome" идентичные, но с точки зрения формализации - нет. Общее в них можно найти взглянув на генерализацию различающейся части. Firefox и Chrome являются пакетами программного обеспечения.

Глава 4

ТЕОРЕТИЧЕСКОЕ ОБОСНОВАНИЕ СИСТЕМЫ, АРХИТЕКТУРА И РЕАЛИЗАЦИЯ

4.1. Модель мышления Марвина Мински

4.1.1. Крити-Селектор-Путь мышления

В 2006 году Марвин Мински опубликовал свою книгу "The emotion machine" [3], 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 4.1 представлена схематичное изображение Критика-Селектора-Пути мышления

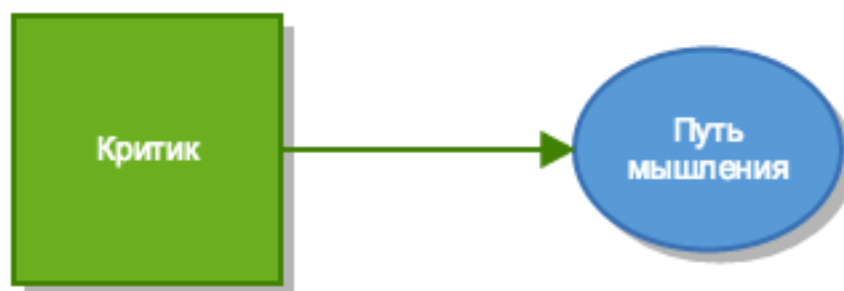


Рисунок 4.1: Критик-Селектор-Путь мышления

Критик 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 занимается выбором определенных ресурсов, которыми также являются Пути мышления.

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

На рисунке 4.2 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.

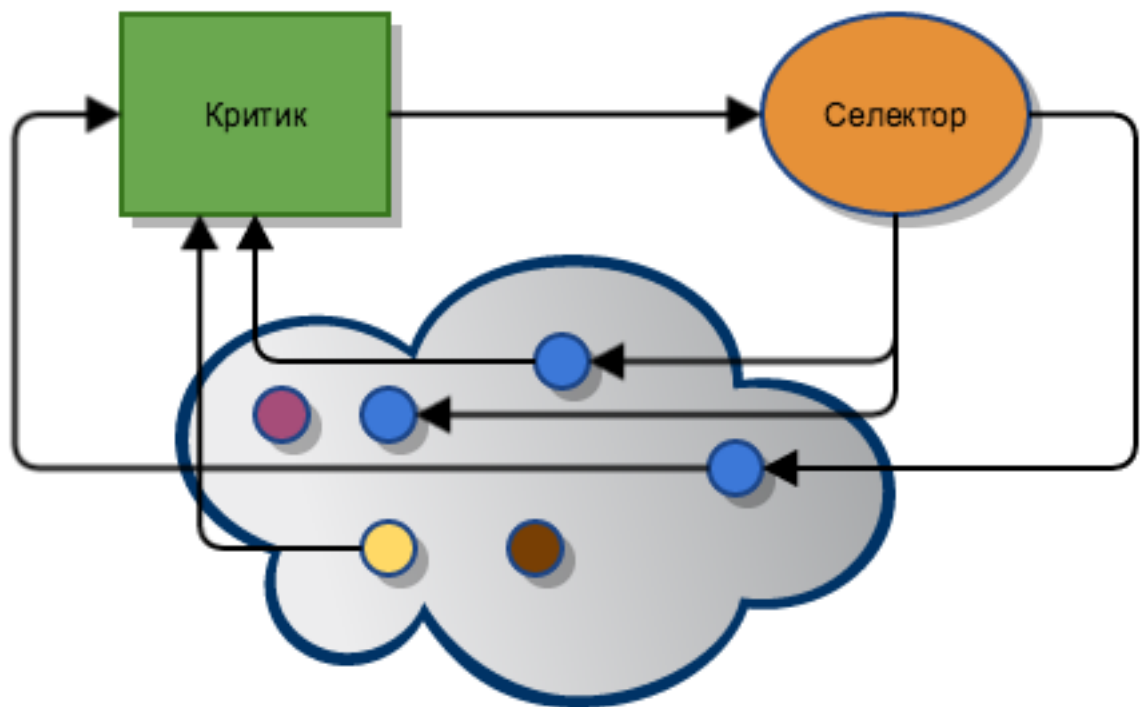


Рисунок 4.2: Критик-Селектор-Путь мышления в разрезе ресурсов

4.1.2. Уровни мышления

Концепция уровней мышления представляет собой степень ментальной активности человека. Никто из людей не может похвастаться скоростью гепарда, гибкостью кошки, силой медведя. Но наш вид все это компенсирует возможностью изобретения путей мышления. Например, чтобы

быть быстрыми мы изобрели самолеты, машины. Чтобы быть сильными, мы изобрели оружие. Что же делает это возможным? Безусловно результатом всего является взаимодействие человека с окружающим миром. Именно данное взаимодействие заставляет людей изобретать что-то новое, создавать шедевры литературы и летать в космос. Но как же мы всего этого добиваемся начиная от инстинктивного одергивая руки до создания Теории всего [14]. Далее мы рассмотрим концепцию уровней мышления.

1. Инстинктивный уровень
2. Уровень обученных реакций
3. Уровень рассуждений
4. Рефлексивный уровень
5. Саморефлексивный уровень
6. Самосознательный уровень

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 данном 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. На данном 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. Самосознательный 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?

Деление на данные уровни носит условный характер. Например уровни 5 и 6 можно объединить. Но по словам Марвина Мински принцип бритвы Оккама успешно применяется в физики, но в психологии он не должен применяться также легко.

На рисунке 4.3 представлена схематичное изображение уровней мышления. 1-3 уровни составляют личность человека. 2-5 представляют ЭГО человека (Человеческое Я) - осознание человека в

общении с окружающими. 3-6 представляют собой сверх ЭГО человека (сверх Я) - его моральные установки.



Рисунок 4.3: Критик-Селектор-Путь мышления в разрезе ресурсов

Заключение

Основные результаты работы заключаются в следующем.

1. На основе анализа ...
2. Численные исследования показали, что ...
3. Математическое моделирование показало ...
4. Для выполнения поставленных задач был создан ...

И какая-нибудь заключающая фраза.

Список литературы

1. *Wikipedia*. IBM Watson. — web. — 2014. https://ru.wikipedia.org/wiki/IBM_Watson.
2. *Wolfram*. Wolfram Alpha. — web. — 2014. — 00. <http://www.wolframalpha.com/>.
3. *Minsky Marvin*. The Emotion Machine. — Simon & Schuster, 2006.
4. *Сычёв М. С.* История Астраханского казачьего войска: учебное пособие. — Астрахань: Волга, 2009. — 231 с.
5. *Соколов А. Н., Сердобинцев К. С.* Гражданское общество: проблемы формирования и развития (философский и юридический аспекты): монография / Под ред. Х. Шутзе. — Астрахань: Издательство, 2009. — 218 с.
6. *Гайдаенко Т. А.* Маркетинговое управление: принципы управленческих решений и российская практика. — 3-е изд, перераб. и доп. изд. — М.: Эксмо: МИРБИС, 2008. — 508 с.
7. *Лермонтов Михаил Юрьевич*. Собрание сочинений: в 4 т. — М.: Терра-Кн. клуб, 2009. — 4 т.
8. Управление бизнесом: сборник статей. — Нижний новгород: Изд-во Нижегородского университета, 2009. — 243 с.
9. *Foundation Apache Software*. Apache OpenNLP. — web. — 2012. — 04. <https://opennlp.apache.org/>.
10. *Goetzel Ben*. OpenCog RelEx. — web. — 2012. — 04. <http://wiki.opencog.org/w/RelEx>.
11. *Вебер П., Вилльямс Д.* Введение в обработку информации / Ed. by Т. Зителло. — Upper Saddle River, New Jersey 07458: Прентис Холл, 2009. — 581 pp.
12. *Гринберг Д.* Надежный алгоритм обработки для грамматики // *Технический отчет Университета Карнеги Мелон CMU-CS-95-125*. — 1995. — 1 июля.
13. *Соколов А. Н., Сердобинцев К. С.* HP OpenView System Administration Handbook: Network Node Manager, Customer Views, Service Information Portal, HP OpenView Operations / Ed. by Х. Шутзе. — Астрахань: Издательство, 2004. — 688 pp.

14. *Хокинг С.* ТЕОРИЯ ВСЕГО. — Москва: Амфора, 2009. — 160 с.

Список рисунков

1.1	Диаграмма состава команд	7
1.2	Диаграмма соотношений типов проблем	7
2.1	Результаты обработки текстов	12
2.2	Архитектура предварительной обработки текста	13
3.1	HP OpenView	17
3.2	Service NOW	18
4.1	Критик-Селектор-Путь мышления	19
4.2	Критик-Селектор-Путь мышления в разрезе ресурсов	20
4.3	Критик-Селектор-Путь мышления в разрезе ресурсов	22

Список таблиц

1.1 Категории инцидентов	8
2.1 Таблица метрик	11

Приложение А

Название первого приложения

Некоторый текст.

Приложение В

Очень длинное название второго приложения, в котором продемонстрирована работа с длинными таблицами

В.1. Подраздел приложения

Вот размещается длинная таблица:

Параметр	Умолч.	Тип	Описание
&INP			
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
продолжение следует			

(продолжение)			
Параметр	Умолч.	Тип	Описание
mars kick	0 1	int int	1: инициализация модели для планеты Марс 0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars kick	0 1	int int	1: инициализация модели для планеты Марс 0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars kick	0 1	int int	1: инициализация модели для планеты Марс 0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars kick	0 1	int int	1: инициализация модели для планеты Марс 0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars kick	0 1	int int	1: инициализация модели для планеты Марс 0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars kick	0 1	int int	1: инициализация модели для планеты Марс 0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars kick	0 1	int int	1: инициализация модели для планеты Марс 0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars kick	0 1	int int	1: инициализация модели для планеты Марс 0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars kick	0 1	int int	1: инициализация модели для планеты Марс 0: инициализация без шума ($p_s = const$) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
&SURFPAR			
kick	1	int	0: инициализация без шума ($p_s = const$)
продолжение следует			

(продолжение)			
Параметр	Умолч.	Тип	Описание
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars kick	0	int	1: инициализация модели для планеты Марс
	1	int	0: инициализация без шума ($p_s = const$)
mars kick	0	int	1: генерация белого шума
	1	int	2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс

В.2. Ещё один подраздел приложения

Нужно больше подразделов приложения!

В.3. Очередной подраздел приложения

Нужно больше подразделов приложения!

В.4. И ещё один подраздел приложения

Нужно больше подразделов приложения!