



Общество с ограниченной ответственностью «САЙТЭК»

Адрес: 115191, Москва, Холодильный пер., д.3, корп.1, стр.4

Тел.: +7 (495) 955-2825, факс: +7 (495) 955-2833

Web: www.sytech.ru, e-mail: info@sytech.ru

ОПИСАНИЕ ПРОГРАММНОГО ПРОДУКТА

Лингвистический процессор «АРИОН-Лингво»

на 86 листах

Москва, 2010г.

СОДЕРЖАНИЕ

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ	4
ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ	5
1 ОБЩИЕ СВЕДЕНИЯ.....	6
1.1 Полное наименование	6
1.2 Цели, назначение и области использования.....	6
2 ОПИСАНИЕ ПРОЦЕССА ДЕЯТЕЛЬНОСТИ	7
2.1 Прикладное описание процесса деятельности.....	7
2.2 Техническое описание процесса деятельности.....	8
2.2.1 Модель предметной области.....	8
2.2.2 Входные данные.....	15
2.2.3 Выходные данные	16
2.2.4 Настройки лингвистического процессора.....	18
3 ОСНОВНЫЕ ТЕХНИЧЕСКИЕ РЕШЕНИЯ.....	21
3.1 Решения по структуре системы, средствам и способам связи для информационного обмена между компонентами системы	21
3.1.1 Сервер обработки неструктурированной информации.....	21
3.1.2 Агент-планировщик.....	31
3.2 Решения по взаимосвязям со смежными системами, обеспечению совместимости	33
3.2.1 Входные данные.....	33
3.2.2 Выходные данные	42
3.3 Решения по режимам функционирования, диагностированию работы системы	50
3.3.1 Рабочий режим	50
3.3.2 Режим конфигурирования.....	50
3.3.3 Режим диагностирования	50
3.4 Решения по численности, квалификации и функциям персонала, режимам его работы, порядку взаимодействия	51
3.4.1 Технический администратор	51
3.4.2 Инженер по знаниям.....	51
3.4.3 Лингвист	52
3.5 Сведения об обеспечении показателей качества	53
3.6 Состав функций, комплексов задач, реализуемых ЛП.....	55
3.6.1 Структурирование естественных языковых текстов	55
3.6.2 Генерация правил по шаблону.....	57
3.6.3 Линеаризация (упрощение естественных языковых текстов).....	57

3.7	Решения по составу информации, объему, способам ее организации	59
3.7.1	Конфигурационные файлы лингвистического процессора	59
3.7.2	Конфигурационные файлы предметных областей	63
3.8	Решения по составу программных средств	74
3.8.1	Особенности обработки русскоязычных текстов	74
4	МЕРОПРИЯТИЯ ПО ПОДГОТОВКЕ К ВВОДУ В ДЕЙСТВИЕ	78
4.1	Мероприятия по приведению информации к виду, пригодному для обработки	78
4.2	Мероприятия по обучению и проверке квалификации персонала	78
4.3	Мероприятия по созданию необходимых подразделений и рабочих мест	79
ПРИЛОЖЕНИЕ 1 ОПИСАНИЕ МОДУЛЯ ПАКЕТНОЙ ОБРАБОТКИ ВХОДНЫХ ДАННЫХ DPS		80
Общие сведения		80
Параметры модуля		80
Параметры конфигурации		81
ПРИЛОЖЕНИЕ 2. ОПИСАНИЕ МОДУЛЯ ТЕСТИРОВАНИЯ ПРАВИЛ ПРЕДМЕТНОЙ ОБЛАСТИ TFW		84
Общие сведения		84
Общая схема работы		84
Параметры конфигурации		84
Результаты работы модуля		85
Состав модуля		86

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Термины

Определения

- | | |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Автоматизированная система | – система, состоящая из персонала и комплекса средств автоматизации его деятельности, реализующая информационную технологию выполнения установленных функций. |
| Анафория | – особенность естественного языка, заключающаяся в пропуске отдельных существительных в тексте либо замене их местоимениями. |
| Аннотирование | – процесс формирования краткого описания основного содержания текста – преимущественно с использованием слов и выражений текста. |
| Данные | – информация, представленная в виде, пригодном для обработки автоматическими средствами при возможном участии человека. |
| Документ | – информационный объект в виде текста, звукозаписи или изображения, размещенный на материальном носителе информации. |
| Идентификация | – проверка информационных объектов на совпадение (похожесть). |
| Информация | – сведения (сообщения, данные) независимо от формы их представления. |
| Информационная система | – совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств. |
| Морфологический анализ | – выделение из текстов на естественном языке лексем, а также синтез словоформ текстов на основе наборов лексем и их морфологических форм. |
| Неструктурированная информация | – информация в виде текста на естественном языке, непригодная для автоматической машинной обработки. |
| Нормализация | – приведение данных к виду, доступному для последующей автоматической обработки. |
| Окраска | – семантический подтип объекта, связи, понятия, лексической единицы. |
| Омонимия | – явление естественного языка, заключающееся в возможности посимвольного совпадения отдельных морфологических форм различных лексических единиц. |
| Предметная область | – фрагмент реального мира, данные о котором хранятся и обрабатываются в информационной системе. |

- Семантическая сеть – способ представления информации в виде ориентированного графа – набора вершин (понятий, объектов), соединенных ребрами (отношениями, связями), т.е. модель предметной области в виде совокупности объектов и связей между ними.
- Синтаксический анализ – выделение синтаксических структур сообщения (текста) на естественном языке на основе соответствующих наборов лексем и их морфологических форм.
- Структурированная информация – форма представления информации о предметной области, построенная в соответствии с логической моделью предметной области, при которой каждый информационный блок однозначно соответствует одному из свойств или отношений отдельных объектов предметной области.
- Тезаурус – способ представления информации о взаимосвязях терминов предметной области в виде перечня лексем с указанием основных лексических отношений между ними.
- Тональность – (текста по отношению к объекту) – характеристика содержания документа или его части как негативного, нейтрального или позитивного по отношению к объекту.
- Целостность информации – способность информационной системы обеспечивать непротиворечивость структурированной информации и её защиту от случайных искажений.

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

- ОС – операционная система
- ПО – программное обеспечение
- FTP – File Transfer Protocol
- HTTP – HyperText Transfer Protocol
- TCP/IP – Transfer Control Protocol / Internet Protocol
- URL – Uniform Resource Locator
- XML – Extensible Markup Language
- XSL – Extensible Style sheet Language

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Полное наименование

Лингвистический процессор «АРИОН-Лингво» (далее – ЛП «АРИОН-Лингво»).

1.2 Цели, назначение и области использования

Лингвистический процессор «АРИОН-Лингво» разрабатывается в целях:

1. Создания универсального программного средства для автоматизированного преобразования неформализованной текстовой информации к структурированному виду на основе predetermined правил.
2. Формирования технологической платформы для создания информационно-аналитических и поисковых систем, предназначенных для автоматизированной обработки значительных массивов неструктурированной текстовой информации.

Лингвистический процессор «АРИОН-Лингво» может использоваться в составе следующих программных продуктов:

1. Информационно-аналитические системы.
2. Поисковые системы и сервисы.

2 ОПИСАНИЕ ПРОЦЕССА ДЕЯТЕЛЬНОСТИ

2.1 Прикладное описание процесса деятельности

Лингвистический процессор «АРИОН-Лингво» предназначен для автоматизированного преобразования неформализованной текстовой информации к структурированному виду в соответствии с параметрами (правилами, словарями и т.д.), определяемыми особенностями предметной области.

В основе лингвистического процессора лежит модель предметной области на основе неоднородной семантической сети. Входные данные последовательно преобразуются в элементы семантической сети (термины, синтаксические структуры, объекты, связи, значения атрибутов) в соответствии с predetermined правилами.

Правила преобразования данных задаются на специализированном непроцедурном языке и интерпретируются в процессе работы ЛП. Правила преобразования данных разбиты на логические слои. На различных слоях выполняются различные операции по обработке данных. Результаты обработки данных на каждом слое подвергаются дополнительной фильтрации для сокращения пространства перебора.

Входными данными для функционирования ЛП «АРИОН-Лингво» являются:

1. Неструктурированные полнотекстовые документы.
2. Частично структурированные документы, содержащие отдельные полнотекстовые поля.
3. DOM-документы, содержащие неструктурированные полнотекстовые фрагменты, а также элементы их разметки и форматирования.

Выходными данными (результатами работы) ЛП «АРИОН-Лингво» являются:

1. Результирующее множество – структурированное описание информационных объектов, значений их атрибутов, связей между объектами, а также соответствия элементов сети исходным данным.
2. Протоколы функционирования ЛП, в которые включаются сведения о процессе обработки неструктурированной информации, полученных результатах и возникающих ошибках.

Необходимыми параметрами для функционирования ЛП «АРИОН-Лингво» являются:

1. Конфигурационные файлы, определяющие общие параметры функционирования и взаимодействия структурных компонентов ЛП.
2. Конфигурационные файлы, определяющие характеристики конкретной предметной области.

2.2 Техническое описание процесса деятельности

2.2.1 Модель предметной области

2.2.1.1 Элементы семантической сети

В основе модели представления данных о предметной области, применяемой в ЛП «АРИОН-Лингво», лежит неоднородная семантическая сеть. Неоднородность сети заключается в том, что ее элементы принадлежат разным классам; каждый класс подразумевает специфический набор характеристик и способов обработки, а также содержательную интерпретацию вне зависимости от конкретной предметной области.

Входные текстовые данные последовательно преобразуются в элементы семантической сети, которые в дальнейшем модифицируются, объединяются и в итоге формируют выходные данные в виде семантической сети, принятой в качестве модели предметной области. Связь элементов семантической сети с исходным текстом сохраняется в характеристиках элемента и используется в дальнейшем (см. 2.2.1.3).

В общем случае элемент семантической сети определяется совокупностью следующих характеристик:

- идентификатор;
- класс элемента семантической сети;
- тип элемента семантической сети (вид элемента внутри класса, определяется предметной областью);
- отображение на лексическую структуру текста;
- приоритет – целочисленное значение, используемое для отбора (фильтрации) элементов;
- значение (окраска) элемента – одно или несколько текстовых описаний элемента;
- идентификаторы связываемых элементов (только для связей);
- атрибуты – список пар «текстовое название – текстовое значение»;
- тона – список троек «текстовое название – текстовое значение – вес» (только для объектов);
- значимость – количество неявных ссылок на этот элемент в обрабатываемом тексте (только для объектов).

Названия атрибутов и тонов элемента семантической сети должны быть уникальными для этого элемента. Атрибуты и тона с пустым названием недопустимы.

В зависимости от конкретного класса и типа элемент семантической сети может описываться различными наборами и значениями характеристик. Кроме того, на разных этапах обработки элемент семантической сети может обладать различными актуальными наборами характеристик, например, значимость и тона присваиваются элементу только на последнем этапе обработки, перед выводом результатов.

Характеристики отображаются в текстовом представлении элемента семантической сети (например, в протоколе):

```
ЮРИДИЧЕСКОЕ_ЛИЦО (%13:15%, 36, АКБ "СВЯЗЬ-БАНК", 392,  
[object])
```

НАИМЕНОВАНИЕ=СВЯЗЬ-БАНК

ФОРМА_СОБСТВЕННОСТИ=АКБ

Здесь, например, «АКБ "СВЯЗЬ-БАНК"» – значение элемента семантической сети, 392 – идентификатор элемента, НАИМЕНОВАНИЕ и ФОРМА_СОБСТВЕННОСТИ – атрибуты элемента, СВЯЗЬ-БАНК и АКБ – значения этих атрибутов, [object] – семантический класс элемента (остальные характеристики более подробно рассмотрены в соответствующих разделах настоящего документа)

2.2.1.2 Семантические классы

Каждый элемент семантической сети относится ровно к одному из следующих семантических классов:

- объект (семантический объект);
- связь (семантическая связь);
- маркер;
- маркерная связь.

2.2.1.2.1 Объекты

Объекты (семантические объекты) образуют класс элементов семантической сети, предназначенных для описания реальных объектов предметной области. Семантический объект определяется следующими характеристиками:

- идентификатор;
- тип семантического объекта (вид объекта с точки зрения информационной модели предметной области);
- отображение объекта на лексическую структуру текста;
- приоритет объекта;
- значение объекта (одно или несколько строковых значений);
- атрибуты объекта (пары «текстовое название – текстовое значение»);
- тона объекта (тройки «текстовое название – текстовое значение – вес»);
- значимость объекта.

При определении предметной области класс семантических объектов подразделяется на типы – виды объектов, различаемые с точки зрения информационной модели этой предметной области. Каждому типу соответствует свой набор обязательных атрибутов и допустимых связей, который определяется в файле `result_types.xml` при описании предметной области (см. п. 2.2.4.2.2).

Например, для одной предметной области в семантическом классе объектов могут быть выделены типы объектов ФИЗИЧЕСКОЕ_ЛИЦО, ОРГАНИЗАЦИЯ, ДОКУМЕНТ, а для другой – типы объектов БАНКОВСКИЙ_СЧЕТ, ОПЕРАЦИЯ, КЛИЕНТ с соответствующими наборами атрибутов и допустимых связей.

Пример отображения семантического объекта в протоколе (файл `out_debug`):

```
ГЕОГРАФИЧЕСКИЙ_ОБЪЕКТ (%2:4;8:10%, 999, МАДРИД, ГОРОД, 1283,  
[object])
```

```
ЧИСЛЕННОСТЬ_НАСЕЛЕНИЯ=1500000
```

Здесь ГЕОГРАФИЧЕСКИЙ_ОБЪЕКТ – тип семантического объекта; «%2:4;8:10%» – маска для связи с текстом (см. п. 2.2.1.3), 999 – приоритет, МАДРИД, ГОРОД – значения, 1283 – идентификатор объекта в семантической сети, [object] – класс объекта как элемента семантической сети, ЧИСЛЕННОСТЬ_НАСЕЛЕНИЯ – атрибут, 1500000 – значение атрибута.

На последней стадии обработки семантическому объекту могут приписываться тона – специальным образом определенные атрибуты, которым сопоставлено числовое значение (вес), и значимость – количество неявных ссылок на объект в обрабатываемом тексте.

2.2.1.2.2Связи

Связи (семантические связи) образуют класс элементов семантической сети, предназначенный для описания отношений между значимыми объектами предметной области в форме отношений между семантическими объектами. Связи описываются следующими характеристиками:

- идентификатор;
- тип связи (вид связи с точки зрения информационной модели предметной области);
- отображение связи на лексическую структуру текста;
- приоритет – целочисленное значение для отсева лишних связей;
- окраска (значение) связи;
- идентификаторы связываемых семантических объектов (в заданном порядке).

В рассматриваемой модели предметной области связи могут связывать ровно два объекта. Связь объекта с самим собой недопустима. Связи в семантической сети всегда считаются направленными от первого из связываемых объектов ко второму. Связи не имеют атрибутов. Связи между связями недопустимы.

Семантика связи определяется ее типом, однако связь может иметь уточняющую окраску – значение, глагол (verb). Возможные типы связей определяются в файле result_types.xml при описании предметной области (см. п. 2.2.4.2.2), однако связи других типов, созданные в результате выполнения правил, по этому критерию не отсеиваются.

Пример отображения связи в протоколе (файл out_debug):

```
СВЯЗЬ_ОПЕРАЦИЯ_ОРГАНИЗАЦИЯ (%2:10;12:15%, 31, 232, 295,
СУБЪЕКТ_ДЕЙСТВИЯ, 449, [link])
```

Здесь СВЯЗЬ_ОПЕРАЦИЯ_ОРГАНИЗАЦИЯ – тип связи, «%2:10;12:15%» – маска для связи с исходным текстом (см. п. 2.2.1.3), 31 – приоритет связи, 232 – идентификатор первого объекта (источник связи), 295 – идентификатор второго объекта (направление связи), СУБЪЕКТ_ДЕЙСТВИЯ – окраска (verb), 449 – идентификатор связи в семантической сети, [link] – семантический класс связи как элемента семантической сети.

2.2.1.2.3Маркеры

Маркеры образуют класс элементов семантической сети, предназначенных для отражения в сети выводных (т.е. являющихся результатами применения правил) сведений об объектах предметной области. Маркер определяется следующим набором характеристик:

- идентификатор;

- тип маркера (вид маркера с точки зрения информационной модели предметной области);
- отображение на лексическую структуру текста;
- приоритет маркера – числовое значение для отсева лишних маркеров;
- значение (окраска) маркера – одно или несколько строковых значений;
- атрибуты маркера – список пар «текстовое название – текстовое значение».

Приоритет маркера всегда устанавливается неявно при создании маркера и определяется по соответствующему тональному словарю в зависимости от значения создаваемого маркера. Маркеры и маркерные связи отсутствуют в выходной семантической сети, так как при ее формировании они преобразуются в тона семантических объектов.

Пример отображения маркера в протоколе (файл out_debug):

```
МАРКЕР_НАСТРОЕНИЕ (%4:6;8:11;18:24;26:28%, 100, ХОРОШЕЕ, 55, [marker])
```

Здесь МАРКЕР_НАСТРОЕНИЯ – тип маркера, «%4:6;8:11;18:24;26:28%» – маска для связи с исходным текстом, 100 – приоритет маркера, ХОРОШЕЕ – значение (окраска) маркера, 55 – идентификатор маркера в семантической сети, [marker] – класс маркера как объекта семантической сети.

2.2.1.2.4 Маркерные связи

Маркерные связи образуют класс элементов семантической сети, предназначенный для описания отношений между семантическими объектами и маркерами. Маркерные связи описываются следующими характеристиками:

- идентификатор;
- тип маркерной связи (в пределах предметной области допустим ровно один тип маркерной связи);
- приоритет маркерной связи – целочисленное значение для отсева лишних связей;
- идентификатор семантического объекта;
- идентификатор маркера.

Маркерная связь всегда считается направленной от семантического объекта к маркеру. В общем случае любой объект может быть связан с любым маркером (это свойство используется при вычислении тонов объектов). Маркерные связи на лексическую структуру обрабатываемого текста не отображаются.

Пример отображения маркерной связи в протоколе (файл out_debug):

```
СВЯЗЬ_ОБЪЕКТ_МАРКЕР (31, 232, 295, 449, [markerlink])
```

Здесь СВЯЗЬ_ОБЪЕКТ_МАРКЕР – тип маркерной связи (постоянный в рамках предметной области), 31 – приоритет маркерной связи, 232 – идентификатор семантического объекта, 295 – идентификатор маркера, 449 – идентификатор маркерной связи в семантической сети, [markerlink] – семантический класс маркерной связи как элемента семантической сети.

2.2.1.3 Отображение элементов семантической сети на входные данные

В лингвистическом процессоре используются несколько уровней представления и обработки данных:

- обобщенный DOM-документ – общий случай входных данных;
- плоский текст – частный случай входных данных;
- лексическая структура – последовательность лексем, соответствующая входным данным;
- семантическая сеть (внутреннее представление данных, изменяющееся по мере их обработки и соответствующее лексической структуре).

Отдельно от приведенной иерархии формируется синтаксическое дерево текста (как дополнительные данные на уровне лексической структуры). Данные о синтаксической роли лексем могут быть получены алгоритмами преобразования семантической сети путем обращения к лексической структуре текста.

Как следствие, существует несколько отображений между перечисленными уровнями данных:

- отображение текста на DOM;
- отображение элементов лексической структуры на текст;
- отображение элементов семантической сети на лексическую структуру.

Между лексической структурой и синтаксическим деревом текста соответствие устанавливается отдельно на этапе синтаксического анализа и не входит в общую иерархию отображений.

2.2.1.3.1 Отображение текста на DOM

Входные данные в форме DOM-документа представляют собой иерархию узлов (элементов, тэгов), определяющих расположение и форматирование фрагментов текста в этом документе. Каждый узел иерархии описывается идентификатором и хранит информацию о структуре документа (параметры тэга, вложенные узлы), а конечные (терминальные) узлы, кроме того, содержат фрагменты текста документа. Отображение текста на DOM определяется ЛП «АРИОН-Лингво» однократно в начале обработки входных данных и в дальнейшем не изменяется.

Для отображения текста документа на DOM группы соседних конечных узлов иерархии (потомков одного узла более высокого уровня) объединяются в секции посредством эвристического алгоритма, основанного на сходстве стиля и расположения соответствующих этим узлам фрагментов исходного документа (computed style).

Таким образом, отображение текста на DOM включает два уровня:

- разбиение плоского текста DOM-документа (совокупности фрагментов, соответствующих всем конечным узлам) на секции;
- соответствие между секциями и конкретными структурными элементами (узлами) DOM.

В дальнейшем сформированный на основе DOM плоский текст обрабатывается лингвистическим процессором целиком, а его отображение на DOM используется правилами преобразования семантической сети и применяется при описании связи элементов семантической сети с входным DOM-документом. В случае, когда исходный документ представлен в форме DOM, такое соответствие задается за счет использования двойной адресации: вначале указывается идентификатор узла DOM, с текстом которого

связывается элемент семантической сети, а затем – смещение и длина одного или нескольких экстенгов (отрезков) текста узла DOM, точно соответствующих этому элементу. Если текст, соответствующий элементу семантической сети, располагается в нескольких узлах DOM, указывается несколько адресующих структур.

2.2.1.3.2 Отображение лексической структуры на текст

Лексическая структура текста представляет собой последовательность лексем исходного текста, соответствующую последовательности слов, знаков препинания, специальных символов и других элементов в тексте. Промежуточной структурой, используемой при отображении лексической структуры на текст, является графематическая структура, включающая все значимые символы и последовательности символов текста (графемы), а также их типы (расширенный набор лексических типов). Отображение лексической структуры на текст строится однократно в процессе графематического и лексического анализа и в дальнейшем не изменяется.

Возможные типы лексем:

- слово из символов кириллического алфавита;
- слово из символов латинского алфавита;
- слово из символов кириллического и латинского алфавитов;
- произвольное слово в кавычках;
- знак препинания;
- блок чисел;
- буквенно-цифровой блок;
- специальный знак (перенос слова, перевод строки, конец предложения).

Все лексемы имеют длину и приоритет, равные 1 (кроме лексемы «специальный знак», длина которой считается равной нулю). Кроме того, лексеме типа «слово» может быть приписан атрибут регистра (верхний, нижний, с заглавной буквой, смешанный).

Отображение лексической структуры на текст состоит в указании для каждого элемента лексической структуры (лексемы) экстенга (отрезка) текста, который соответствует этой лексеме. Такое соответствие задается смещением экстенга относительно начала текста и его длиной (в символах).

Отображение лексической структуры на текст используется правилами преобразования семантической сети (например, для определения расстояний между упоминанием объектов в тексте) и применяется при формировании выходных данных для описания связи элементов семантической сети с исходным текстом. Соответствие между элементом семантической сети и описывающим его текстом устанавливается через лексическую структуру.

2.2.1.3.3 Отображение элементов семантической сети на лексическую структуру

Отображение элементов семантической сети на лексическую структуру текста определяется ЛП при формировании начальной семантической сети и в дальнейшем модифицируется по мере обработки и модификации сети.

Каждый элемент семантической сети, относящийся к классам семантических объектов, семантических связей и маркеров, соответствует одной или нескольким лексемам в лексической структуре обрабатываемого текста (т.е. одному или нескольким словам текста). При этом для семантических объектов возможны два варианта соответствия:

- семантический объект соответствует одной лексеме либо нескольким лексемам, образующим непрерывный фрагмент лексической структуры (непрерывный объект);
- семантический объект соответствует нескольким лексемам, между которыми в лексической структуре текста присутствуют лексемы, не относящиеся к этому объекту (разрывный объект).

Заметим, что признак разрывности/непрерывности по аналогии может применяться к семантическим связям и маркерам, однако последующая классификация опирается на это свойство только применительно к семантическим объектам.

Непрерывные объекты образуют начальную семантическую сеть, формируемую на основе лексической структуры текста, а также семантическую сеть, являющуюся результатом применения к начальной сети правил первого слоя. Это объясняется тем, что в целях повышения производительности интерпретатор применяет правила первого слоя только к объектам и лексемам, соответствующим непрерывным фрагментам лексической структуры (подробнее см. п. 3.7.2.4).

Отображение элемента семантической сети на лексическую структуру текста определяется последовательностью фрагментов лексической структуры, соответствующих этому элементу. Формат описания фрагмента лексической структуры включает индексы первой и последней лексем фрагмента в лексической структуре текста, разделённые двоеточием. Последовательность фрагментов ограничивается символами «%», а её элементы разделяются точкой с запятой.

Отображение элементов семантической сети на лексическую структуру текста явно присутствует в протоколе обработки входных данных (файл `out_debug`) и неявно применяется при указании соответствия элементов результирующей семантической сети входным данным.

2.2.1.3.4 Отображение синтаксического дерева на лексическую структуру текста

Синтаксическое дерево текста формируется ЛП в процессе синтаксического анализа текста и в дальнейшем не изменяется. Отображение синтаксического дерева на текст не входит в иерархию отображений, связывающих элементы семантической сети с элементами DOM исходного документа. Тем не менее, данные о синтаксической роли лексемы сохраняются ЛП и могут быть использованы правилами обработки семантической сети при обращении к этой лексеме.

Элементами синтаксического дерева текста являются синтаксические структуры – непрерывные последовательности синтаксически связанных или подчиненных лексем. Синтаксическое дерево представляет собой иерархию синтаксических структур, описывающую текст в целом, причем синтаксические структуры, соответствующие дочерним узлам дерева, полностью покрываются синтаксической структурой, соответствующей родительскому узлу.

Типы синтаксических структур:

- текст;
- параграф;
- предложение;
- клауза (часть сложного предложения);
- группа (словосочетание).

Синтаксическая структура «текст» соответствует корню синтаксического дерева. Структура каждого последующего типа может включаться только в структуру предыдущего типа (за исключением клауз, которые могут включать другие клаузы). Синтаксические структуры «группа» и «клауза» относятся к одному из predetermined типов (см. п. 3.8.1).

Каждая синтаксическая структура соответствует некоторому фрагменту лексической структуры. Соответствие определяется индексом первой лексемы, соответствующей этой синтаксической структуре, в лексической структуре текста и количеством лексем, относящихся к этой синтаксической структуре.

2.2.2 Входные данные

ЛП «АРИОН-Лингво» выполняет обработку следующих видов входных данных:

- плоский текст на русском или английском языке (информационное сообщение);
- частично формализованные документы – специализированные файлы формата XML, содержащие как структурированные описания объектов, так и неформализованные тексты в форме полнотекстовых атрибутов этих объектов;
- DOM-документы – файлы формата XML, описывающие DOM-модель, полученную в результате анализа структуры HTML-документа.

2.2.2.1 Текстовые входные данные

Плоские тексты (неструктурированные полнотекстовые документы, информационные сообщения) представляют собой наиболее активно используемый класс входных данных ЛП.

Тексты могут быть представлены на русском или английском языках в произвольной кодировке. На этапе препроцессирования тексты автоматически преобразуются в кодировку CP-1251, при этом из них удаляются недопустимые (необрабатываемые) символы. ЛП в автоматическом режиме заменяет последовательности символов в тексте в соответствии со словарями соответствующей предметной области, что позволяет расшифровывать аббревиатуры и исправлять типичные орфографические ошибки.

Для каждого текста указывается текстовый идентификатор предметной области, параметры которой (правилами, словарями, перечнем типов объектов и т.д.) определяют особенности его обработки на всех этапах. В результате разбора текста строится единая результирующая семантическая сеть.

Более подробно текстовые входные данные рассмотрены в п. 3.2.1.1.

2.2.2.2 Частично структурированные документы

Частично структурированные документы представляют собой XML-файлы, содержащие описания информационных объектов, значений их атрибутов и связей между объектами. Лингвистическим процессором в этом случае обрабатываются полнотекстовые атрибуты объектов.

Для каждого информационного объекта может быть указан текстовый идентификатор предметной области, правила которой применяются для обработки полнотекстовых атрибутов этого объекта. Таким образом, каждый объект может обрабатываться ЛП в соответствии с параметрами отдельной предметной области. Вне зависимости от указанной предметной области каждый атрибут обрабатывается отдельно как входные данные вида «плоский текст».

Формат представления частично структурированных входных данных рассмотрен в п. 3.2.1.2.

2.2.2.3 DOM-документы

DOM-документы представляют собой XML-файлы, определяющие структуру размеченного текста (т.е. разбиение его на отдельные структурные элементы, такие, как сообщения, их тексты и заголовки, иллюстрации к сообщениям и т.д.). DOM-документы формируются автоматически в результате обработки HTML-документа или его фрагмента специализированным плагином браузера Internet Explorer.

Лингвистический процессор разбивает текст DOM-документа на секции (последовательности соседних структурных элементов) и обрабатывает его как входные данные вида «плоский текст». Предметная область определяется одним текстовым идентификатором для всего DOM-документа. Текст DOM-документа обрабатывается целиком, а сведения о его разбиении на секции учитываются на этапе фрагментации (см. п. 3.1.1.4.3).

Формат представления DOM-документа рассмотрен в п. 3.2.1.3.

2.2.3 Выходные данные

К выходным данным ЛП «АРИОН-Лингво» относятся:

- результирующее множество;
- протоколы обработки входных данных.

2.2.3.1 Результирующее множество

Результирующее множество – структурированное описание информационных объектов – является основным видом выходных данных и представляет собой файл формата XML, содержащий описания семантических объектов, значений атрибутов объектов, связей между объектами, а также сведений о соответствии этих объектов и связей лексической и синтаксической структуре входных данных. Сами входные данные в результирующем множестве не повторяются.

Результирующее множество представляет собой выборку из финальной семантической сети, полученной в результате обработки входных данных, содержащую только элементы классов «семантический объект» и «семантическая связь», соответствующие набору типов, допустимых с точки зрения предметной области. Для преобразования финальной семантической сети в результирующее множество на последнем этапе работы ЛП выполняются следующие операции:

- преобразование элементов семантической сети классов «маркер» и «маркерная связь» в взвешенные атрибуты (тона) семантических объектов. Таким образом, в результирующее множество включаются только элементы сети, непосредственно связанные с входными данными;
- отсеивание семантических объектов и связей, не соответствующих перечню типов объектов предметной области. Таким образом, отсеиваются служебные и временные объекты и связи, возникающие в процессе выполнения правил.

Результирующее множество в зависимости от вида входных данных представляется в одном из двух форматов.

Базовый формат результирующего множества (см. п. 3.2.2.1.1) используется для представления результатов обработки ЛП входных данных, представленных в виде плоского текста или DOM-документа, и включает следующие данные:

- семантические объекты и значения их атрибутов;
- связи между семантическими объектами;
- соответствие семантических объектов и связей исходному тексту;
- синтаксическое дерево исходного текста.

Для указания соответствия семантических объектов и связей исходному тексту применяется одноуровневая адресация отрезков текста. Каждому объекту соответствует один или несколько отрезков текста (экстентов), задаваемых смещением отрезка относительно начала исходного текста и длиной отрезка.

Формат частично структурированного документа (см. п. 3.2.2.1.2) применяется для представления результатов обработки ЛП входных данных, представленных также в виде частично структурированного документа, и включает следующие данные:

- исходные информационные объекты, их связи и атрибуты (в том числе – текстовые поля);
- семантические объекты, их атрибуты и связи (результаты работы ЛП);
- соответствие семантических объектов и связей текстовым полям исходного документа;
- элементы синтаксической структуры исходных текстов.

Во входных данных текстовые поля представляют собой символьные строки со ссылками на информационные объекты, к которым они относятся. Для указания соответствия семантических объектов и связей, полученных в результате работы ЛП, текстовым полям исходного документа применяется двухуровневая адресация. Каждому семантическому объекту или связи соответствует один или несколько отрезков (экстентов) текстовых полей исходного документа, задаваемых вначале идентификатором поля, а затем – смещением отрезка относительно начала этого поля и длиной отрезка.

2.2.3.2 Протоколы обработки входных данных

Протоколы обработки входных данных содержат сведения о процессе последовательного преобразования этих исходных данных в элементы семантических сетей (от начальной до результирующей) при помощи предусмотренных правил.

В протоколы ЛП включаются следующие сведения:

- снимки семантической сети между последовательными этапами ее обработки;
- снимки результатов фильтрации семантической сети;
- снимки синтаксических структур, пригодные для графического отображения;
- снимки истории элементов семантической сети.

Помимо перечисленных данных, в протокол включаются статистические данные о функционировании ЛП, временные метки, а также сами входные данные и результирующие множества.

2.2.4 Настройки лингвистического процессора

К настройкам лингвистического процессора относятся:

- конфигурационные файлы ЛП;
- конфигурационные файлы предметных областей ЛП.

2.2.4.1 Конфигурационные файлы лингвистического процессора

Конфигурационные файлы ЛП определяют основные параметры функционирования и взаимодействия его структурных компонентов. Отдельные конфигурационные файлы предусмотрены для следующих компонентов:

- сервер, обеспечивающий непосредственное выполнение запросов по обработке входных данных;
- агент, обеспечивающий управление нагрузкой на серверов и конфигурациями предметных областей.

Конфигурационные файлы содержат следующие основные данные:

- для сервера: технические параметры для связи с агентом, каталоги размещения словарей и правил, ограничения времени ожидания, параметры ведения протокола и т.д.;
- для агента: каталоги размещения описаний предметных областей, технические параметры для связи с сервером, параметры производительности и т.д.

Форматы конфигурационных файлов агента-планировщика и сервера ЛП рассмотрены в п.п. 3.7.1.1 и 3.7.1.2 соответственно.

2.2.4.2 Конфигурационные файлы предметных областей

Настройки предметной области определяются следующими основными видами конфигурационных файлов:

- описание состава конфигурационных файлов;
- общие настройки предметной области;
- словари предметной области;
- правила предметной области;
- манифесты внешних библиотек.

2.2.4.2.1 Описание состава конфигурационных файлов

Корневым файлом настройки предметной области, определяющим состав конфигурационных файлов предметной области, является файл `config_part.xml`, формат которого рассмотрен в п. 3.7.2.1.

2.2.4.2.2 Общие настройки предметной области

К общим настройкам предметной области относятся:

- перечень типов объектов и связей предметной области;
- настройки хэширования семантических объектов.

Перечень типов объектов и связей используется в процессе получения результирующего множества на основе фильтрации семантической сети, при которой отсеиваются семантические объекты, типы которых не соответствуют этому перечню. Семантические связи непосредственно по типам не отсеиваются и удаляются только в тех

случаях, когда нарушаются дополнительные ограничения (например, у некоторого объекта количество связей определенного типа превосходит ограничение на количество связей такого типа, установленное для этого типа объектов).

Для идентификации семантических объектов, типы которых включены в перечень типов объектов и связей предметной области, используются значения хэш-функций, вычисляемых на ключевых наборах атрибутов. Одному типу объектов может быть сопоставлено несколько ключевых наборов атрибутов, и, как следствие, несколько хэш-функций. Таким образом, настройки хэширования семантических объектов определяют состав ключевых наборов атрибутов и идентификаторы соответствующих хэш-функций.

Общие настройки предметной области определяются в файле `result_types.xml`, формат которого рассмотрен в п. 3.7.2.2.

2.2.4.2.3 Словари предметной области

Словари предметной области подразделяются на следующие типы:

- словари транслитераций, предназначенные для транслитерации русскоязычных текстов;
- словари рекодера, содержащие инструкции по замене символов, не представимых в кодировке Windows-1251, другими символами или по их исключению;
- словари расшифровок (аббревиатур), определяющие правила преобразования аббревиатур в развернутые словосочетания;
- словари опечаток, устанавливающие соответствие между ошибочными и правильными вариантами написания слов;
- словари категорий, предназначенные для классификации слова в соответствии с его семантической категорией (например, слово «декабрь» относится к категории «месяц»);
- тональные словари, определяющие категории (типы тональностей) и веса слов внутри категорий;
- универсальные словари, определяющие категории, их коды и иерархии категорий.

Формат представления словарей рассмотрен в п. 3.7.2.3.

2.2.4.2.4 Правила предметной области

Правила предметной области содержат инструкции по преобразованию семантической сети и подразделяются на следующие типы:

- правила нулевого слоя, предназначенные для разбиения исходного текста на фрагменты, обрабатываемые по отдельности в целях повышения производительности;
- правила первого слоя, обеспечивающие создание и модификацию семантических объектов в начальной семантической сети;
- правила второго слоя, предназначенные для формирования связей между объектами семантической сети;
- правила третьего слоя, определяющие инструкции по созданию маркеров для объектов семантической сети;

- правила четвертого слоя, предназначенные для формирования сложных семантических объектов, содержащих несколько определённых ранее объектов и связей.

Формат представления правил обработки приведен в п. 3.7.2.4.

2.2.4.2.5 Манифесты внешних библиотек

Внешние библиотеки представляют собой подключаемые модули ЛП, содержащие прикладные функции, используемые правилами предметной области для проверки условий и манипуляции данными. Манифесты внешних библиотек содержат следующие сведения:

- состав словарей предметной области, используемых прикладными функциями внешней библиотеки.

Формат файла манифеста внешней библиотеки рассмотрен в п. 3.7.2.5.

3 ОСНОВНЫЕ ТЕХНИЧЕСКИЕ РЕШЕНИЯ

3.1 Решения по структуре системы, средствам и способам связи для информационного обмена между компонентами системы

В состав ЛП «АРИОН-Лингво» входят следующие структурные компоненты:

- сервер обработки неструктурированной информации, обеспечивающий преобразование неструктурированных входных данных к формализованному виду;
- агент-планировщик, обеспечивающий управление задачами по обработке информации и контролирующий нагрузку на экземпляры сервера, расположенные на аппаратных средствах.

3.1.1 Сервер обработки неструктурированной информации

Программное обеспечение сервера позволяет выполнять обработку входных данных в соответствии с настройками одной предметной области. Таким образом, для выполнения заданий по обработке текстов, относящихся к различным предметным областям, запускаются несколько экземпляров сервера, каждый из которых сконфигурирован в соответствии с нужной предметной областью. Запуск серверов, направление им запросов клиентов и остановку серверов выполняет агент-планировщик.

С технической точки зрения сервер обработки неструктурированной информации представляет собой многопоточный процесс (служба на платформе Win32 или демон в UNIX-подобных операционных системах). Сервер предоставляет программный интерфейс для асинхронного взаимодействия с агентом-планировщиком по протоколу TCP/IP.

Функциональная структура сервера обработки неструктурированной информации включает следующие подсистемы:

- сетевая подсистема, обеспечивающая управление очередью запросов, ранжирование запросов по приоритетам и управление потоками обработки запросов;
- подсистема предварительной обработки, обеспечивающая приведение входных данных к виду, пригодному для дальнейшей обработки;
- лингвистическая подсистема, обеспечивающая графематический, морфологический и синтаксический анализ обрабатываемых текстов;
- семантическая подсистема, реализующая построение начальной семантической сети, ее последовательное преобразование в соответствии с правилами предметной области и формирование результирующего множества;
- подсистема управления настройкой на предметную область, обеспечивающая доступ других компонентов к перечню типов, словарям, внешним функциям, правилам и хэш-функциям предметной области.

Исходный код сервера обработки неструктурированной информации разработан на языке программирования C++, что обеспечивает его свободную переносимость. В настоящее время поддерживаются гарантированные конфигурации сервера для ОС Windows XP, Windows Server 2003 и выше, для ОС Linux с версией ядра не ниже 2.6 и для Mac OS X версий 10.5 и выше.

3.1.1.1 Сетевая подсистема

Сетевая подсистема сервера обработки неструктурированной информации предназначена для получения от агента-планировщика запросов на обработку текстов и передачи их для обработки другим компонентам сервера. В состав подсистемы входят следующие компоненты:

- модуль управления очередью запросов;
- средство разбора (парсер) транспортного формата.

3.1.1.1.1 Модуль управления очередью запросов

Модуль управления очередью запросов обеспечивает общее управление поступающими запросами и результатами их обработки. К функциям модуля управления очередью запросов относятся:

- получение запроса от агента-планировщика и определение его типа (запрос на обработку данных или диагностический запрос);
- внеочередное выполнение диагностических запросов и передача результатов их выполнения агенту-планировщику;
- постановка запроса на обработку данных в очередь запросов;
- ранжирование очереди запросов в зависимости от их приоритетов;
- извлечение очередного запроса из очереди и передача его потоку обработки данных для выполнения;
- получение результатов обработки запроса от потока обработки данных;
- передача агенту-планировщику результатов выполнения запроса;
- отмена выполнения запроса по инициативе агента-планировщика.

Для обеспечения параллельной обработки запросов в сервере используется пул выполняющихся потоков обработки данных, каждому из которых может быть передан на обработку запрос из очереди. На параллельную обработку запросов накладываются ограничения (например, одновременно могут выполняться запросы на обработку данных и любые диагностические запросы, но не любые запросы на обработку данных могут выполняться одновременно).

Модуль управления очередью запросов функционирует постоянно в процессе работы сервера и является уникальным в рамках одного сервера. Настраиваемыми параметрами модуля являются адрес узла и порт для обращения к серверу агента-планировщика, определяемые в конфигурационном файле сервера (см. п. 3.7.1.2).

3.1.1.1.2 Средство разбора (парсер) транспортного формата

Парсер транспортного формата предназначен для преобразования исходного XML-представления запроса к внутреннему представлению ЛП и используется для обеспечения прозрачного взаимодействия модуля управления очередью запросов с агентом-планировщиком. Парсер транспортного формата выполняет следующие функции:

- извлечение XML-представления запроса из входного TCP-потока;
- формирование внутреннего представления запроса в результате анализа его исходного XML-представления;
- генерация XML-представления результатов обработки запроса;
- помещение XML-представления результатов запроса в выходной TCP-поток.

Парсер транспортного формата является неотключаемым модулем, причем одним сервером используется ровно один парсер. Настраиваемых параметров модуль не имеет.

3.1.1.2 Подсистема предварительной обработки

Подсистема предварительной обработки предназначена для приведения текстовых данных запросов к виду, пригодному для автоматической обработки основными компонентам сервера. В состав подсистемы входят следующие компоненты:

- рекодер;
- препроцессор.

3.1.1.2.1 Рекодер

Рекодер предназначен для преобразования текста запроса к кодировке Windows-1251, являющейся основной внутренней кодировкой ЛП, и выполняет следующие функции:

- загрузка конфигурационного файла рекодера;
- преобразование текста запроса к кодировке Windows-1251.

Рекодер является подключаемым модулем, причем одновременно может использоваться не более одного рекодера для каждой предметной области. Для подключения рекодера в файл конфигурации предметной области `config_part.xml` добавляется параметр

```
<recoder>recoder.xml</recoder>
```

Здесь `recoder.xml` – имя конфигурационного файла рекодера, содержащего инструкции по преобразованию интервалов символов в символы кодировки Windows-1251. Формат конфигурационного файла рекодера рассмотрен в п. 2.2.4.2.

3.1.1.2.2 Препроцессор

Препроцессор предназначен для предварительной обработки лексем, выделенных в тексте запроса, путем их замены в соответствии с предусмотренными инструкциями. К функциям препроцессора относятся:

- загрузка конфигурационного файла препроцессора;
- замена отдельных элементов DOM и лексем в текстовых данных запроса в соответствии с предусмотренными инструкциями.

Препроцессор рассматривает текст как результат графематического анализа и не выполняет разбиения или объединения отдельных лексем. Основной целью работы препроцессора является исправление опечаток оператора и орфографических ошибок до разбиения текста на фрагменты.

Препроцессор является подключаемым модулем, причем одновременно допустимо использование только одного препроцессора для одной предметной области. Для подключения препроцессора в файл конфигурации предметной области `config_part.xml` добавляется параметр

```
<preprocessor>pp.xml</preprocessor>
```

Здесь `pp.xml` – имя конфигурационного файла препроцессора, содержащего инструкции по преобразованию лексем, при этом пустое значение параметра или его отсутствие интерпретируется как отключение препроцессора. Формат конфигурационного файла препроцессора рассмотрен в п. 2.2.4.2.

3.1.1.3 Лингвистическая подсистема

Лингвистическая подсистема сервера обработки неструктурированной информации предназначена для преобразования текста запроса в структуру, содержащую все необходимые данные об элементах текста и их взаимосвязи в тексте. В состав подсистемы входят следующие компоненты:

- графематический анализатор;
- лексический процессор;
- морфологический анализатор;
- синтаксический анализатор.

3.1.1.3.1 Графематический анализатор

Графематический анализатор выполняет разбиение текста запроса на элементы (графемы) в соответствии с разделителями, естественными для языка (пробелами, специальными символами, знаками препинания), причем сами разделители могут интерпретироваться анализатором как новые элементы текста. К функциям графематического анализатора относятся:

- формирование графематической структуры текста (последовательности элементов текста – графем).

Графематическая структура текста является промежуточной структурой между строковым представлением текста и его лексической структурой, поэтому отдельно в рамках используемой модели предметной области (см. п. 2.2.1) не рассматривается. Технически графематическая структура сохраняется и обрабатывается ЛП способом, аналогичным хранению и обработке лексической структуры.

Графематический анализатор функционирует постоянно в процессе работы сервера. Настраиваемым параметром анализатора является язык текста, определяемый в конфигурационном файле сервера (см. п. 3.7.1.2).

3.1.1.3.2 Лексический процессор

Лексический процессор предназначен для подготовки результатов графематического анализа к этапу создания лексем, т.е. выявлению графем и последовательностей графем, относящихся к отдельным лексемам. К функциям лексического процессора относятся:

- определение границ лексем на основе словаря разделителей лексем, в том числе – удаление переносов;
- обработка однословных сокращений (определение ложных границ предложений);
- анализ коротких последовательностей и прямой речи (классификация кавычечных выражений на неанализируемые однолексемные названия и содержательные фрагменты, подлежащие дальнейшей обработке).

Словарь разделителей лексем является элементом настройки на предметную область и задается в файле конфигурации ЛП `config_part.xml` обязательным параметром

```
<delims>  
  <item on="true">delimiters.xml</item>  
</delims>
```

Здесь `delimiters.xml` – имя файла, содержащего словарь разделителей лексем и находящегося в каталоге текущей предметной области; параметр `on` определяет необходимость использования словаря в текущей конфигурации предметной области.

Для анализа коротких последовательностей и прямой речи применяется алгоритм, позволяющий разбить текст со вложенными кавычными выражениями на непротиворечивые фрагменты так, чтобы длина каждого выражения не превышала некоторый порог (обычно 3-5). Работа алгоритма основана на построении дерева экстенгов (наборов фрагментов текста между соседними кавычками) и оптимизационного поиска по нему.

Лексический процессор функционирует постоянно в процессе работы сервера. Формат словарей разделителей лексем рассмотрен в п. 2.2.4.2.3.

3.1.1.3.3 Морфологический анализатор

Морфологический анализатор позволяет определить морфологическую форму отдельного слова или сгенерировать слово по его начальной форме и заданной морфологической форме. К функциям морфологического анализатора относятся:

- определение начальной формы заданного слова;
- получение всех морфологических форм заданного слова;
- генерация согласованных последовательностей морфологических форм нескольких слов (например, словосочетаний).

Результаты морфологического анализа представляются в виде морфологических форм – текстовых строк, содержащих текстовое представление слова и двухсимвольную грамему, однозначно определяющую морфологическую форму этого слова. Морфологическая информация о словах не сохраняется в процессе обработки текста, поэтому компоненты ЛП обращаются к процедуре морфологического анализа по мере необходимости.

Морфологический анализатор является неотключаемым компонентом сервера и использует в процессе работы встроенные морфологические функции языка правил предметной области. К параметрам конфигурации анализатора опосредованно относятся используемые встроенными функциями словари предметной области, формат которых рассмотрен в п. 3.7.2.3.

3.1.1.3.4 Синтаксический анализатор

Синтаксический анализатор позволяет построить синтаксическое дерево текста. Функциями синтаксического анализатора являются:

- разбиение текста на отдельные предложения;
- выделение всех синтаксических структур текста;
- построение синтаксического дерева текста;
- формирование отображения синтаксического дерева текста на лексическую структуру текста.

В процессе синтаксического анализа используются результаты работы лексического процессора в части определения границ предложений, а также процедура морфологического анализа. На основе этих данных формируются варианты синтаксических структур текста и варианты синтаксических деревьев, среди которых выбирается дерево с максимальным приоритетом.

Синтаксический анализатор является неотключаемым компонентом сервера. Настраиваемым параметром конфигурации анализатора является язык обрабатываемого текста (см. п. 3.7.1.2).

3.1.1.4 Семантическая подсистема

Семантическая подсистема предназначена для преобразования текста в начальную семантическую сеть, преобразования сети в соответствии с предусмотренными правилами и формирования результирующего множества элементов семантической сети. В состав семантической подсистемы входят следующие компоненты:

- структуризатор;
- лексический анализатор;
- фрагментатор;
- модуль обработки текстов;
- модуль постобработки.

3.1.1.4.1 Структуризатор

Структуризатор предназначен для разбиения текста DOM-документа на секции в соответствии со структурой документа. К функциям структуризатора относятся:

- определение в иерархической структуре DOM-документа секций (в зависимости от расположения и форматирования текста);
- сохранение данных о разбиении текста на секции для последующей обработки;
- формирование соответствующего отображения текста на DOM.

Секции в DOM выделяются по признакам близости расположения и схожести форматирования (в том числе выявляются однородные и повторяющиеся иерархические последовательности). Полученное разбиение на секции используется правилами предметной области при выделении объектов и связей.

Исходные данные, представленные в виде текста и частично формализованного документа, структуризатором не обрабатываются и передаются для дальнейшей обработки без изменений.

Структуризатор является неотключаемым компонентом сервера и не имеет конфигурируемых параметров.

3.1.1.4.2 Лексический анализатор

Лексический анализатор предназначен для формирования лексической структуры текста на основе его графематической структуры, т.е. сопоставлению каждому значимому элементу текста однозначно соответствующей ему лексеме.

К функциям лексического анализатора относятся:

- идентификация типов графем (слово, иностранное слово, последовательность букв и цифр, слово в кавычках, знак препинания и т.д.);
- построение лексем для графем значимых типов и формирование лексической структуры текста;
- формирование соответствующего отображения лексической структуры на текст (с использованием промежуточной графематической структуры).

Лексический анализатор является неотключаемым компонентом сервера и не имеет конфигурируемых параметров.

3.1.1.4.3 Фрагментатор

Фрагментатор предназначен для разбиения обрабатываемого текста на фрагменты, имеющие меньшую длину, чем исходный текст. В дальнейшем обработка текста выполняется для каждого фрагмента по отдельности, а результаты обработки фрагментов объединяются модулем постобработки. Функциями фрагментатора являются:

- загрузка правил нулевого слоя из конфигурационного файла и формирование динамического интерпретатора линейной структуры (выполняется однократно при инициализации модуля);
- определение возможных границ фрагментов на основе применения правил нулевого слоя и эвристического алгоритма;
- поиск минимального бесконфликтного набора фрагментов, удовлетворяющих всем правилам и эвристикам;
- построение лексических структур каждого фрагмента и отображений этих структур на тексты фрагментов.

Алгоритм работы фрагментатора включает два этапа. На первом этапе применяются правила нулевого слоя, в секции инструкций которых используется оператор АССЕРТ. Семантика правил нулевого слоя состоит в выборе в тексте лексем определенного типа, и, если расстояние между такими лексемами не превышает установленного ограничения, пометки всего интервала текста между этими лексемами как неделимого, т.е. относящегося к одному фрагменту.

На втором этапе работы фрагментатора выполняются попытки разбиения текста на фрагменты по границам предложений и абзацев, при этом текст не разбивается в точках, входящих в неделимые интервалы. Фрагментирование завершается применением адаптивного алгоритма для поиска минимального бесконфликтного набора фрагментов.

Правила нулевого слоя загружаются фрагментатором из файла, название которого указано в соответствующем параметре конфигурации предметной области (см. п. 3.7.2). Для всех правил создаются интерпретирующие объекты, отражающие семантику этих правил, которые объединяются в динамический интерпретатор линейной структуры. Таким образом, в каждый момент времени фрагментатор может быть настроен только на одну предметную область, а для изменения этой настройки необходима его повторная инициализация.

3.1.1.4.4 Модуль обработки текстов

Модуль обработки текстов предназначен для выполнения непосредственной обработки текста в соответствии с настройкой на конкретную предметную область. Модуль обладает внутренней структурой, компонентами которой являются:

- фабрика интерпретаторов – функциональный компонент, обеспечивающий создание интерпретирующего объекта для конкретного правила и размещение его в структуре интерпретатора;
- массив динамических интерпретаторов – массив линейных и иерархических структур, состоящих из интерпретирующих объектов. Каждому слою правил предметной области соответствует динамический интерпретатор.

К функциям фабрики интерпретаторов относятся (эти функции выполняются однократно в процессе инициализации модуля):

- построение на основе загруженных парсером правил преобразования семантической сети динамических интерпретаторов линейной структуры для второго и третьего слоёв;

- построение на основе загруженных парсером правил преобразования семантической сети динамических интерпретаторов иерархической структуры для первого и четвертого слоёв.

К функциям динамических интерпретаторов, выполняемым при обработке каждого текста, относятся:

- построение начальной семантической сети на основе лексической структуры обрабатываемого текста;
- применение правил первого слоя к начальной семантической сети и её модификация в соответствии с этими правилами;
- применение к семантической сети правил второго слоя для создания связей между семантическими объектами;
- применение к семантической сети правил третьего слоя для создания тональных маркеров;
- применение к семантической сети правил четвертого слоя для создания сложных семантических объектов, включающих несколько определенных ранее объектов и связей;
- фильтрация семантической сети, в процессе которой отсеиваются: элементы семантической сети, типы которых не соответствуют семантике предметной области; конфликтные элементы; элементы с низким приоритетом;
- формирование результирующего множества (фильтрация семантической сети и преобразование маркеров и маркерных связей в множественные атрибуты (тона) семантических объектов).

Модуль обработки текстов в каждый момент времени может быть настроен только на одну предметную область, конфигурационные файлы которой загружаются подсистемой управления настройкой на предметную область при инициализации модуля. Для обработки текстов другой предметной области модуль должен быть повторно инициализирован, а его конфигурационные файлы перезагружены.

Построение динамических интерпретаторов осуществляется в три этапа. На первом этапе используется парсер правил, выполняющий разбор соответствующего конфигурационного файла. На втором этапе фабрика интерпретаторов выделяет среди загруженных данных элементы правил (переменные, функции, условия, константы, регулярные выражения, операторы, параметры правил, настройки фильтров и т.д.). На третьем этапе фабрика интерпретаторов генерирует для каждого правила интерпретирующий объект, объединяющий все элементы этого правила, при этом выполняется контроль семантики построенного правила. На этапе генерации интерпретирующего объекта инициализируются и связываются переменные запросов, аргументы функций и операторов.

Интерпретирующие объекты, соответствующие правилам предметной области, объединяются фабрикой в динамические интерпретаторы. Для каждого слоя правил создается интерпретатор (для первого и четвертого слоев он имеет древовидную структуру, для второго и третьего – линейную). Применение правил предметной области к семантической сети реализовано за счёт обработки семантической сети динамическим интерпретатором соответствующего слоя.

Фильтрация элементов семантической сети предназначена для отсева семантических объектов и связей, тип, приоритет или взаимное расположение которых в тексте не соответствует семантике предметной области, и основана на алгоритме построения

бесконфликтной сети с максимальным приоритетом. Например, отсеиваются те элементы, при отображении которых на лексическую структуру текста возникают конфликты с элементами, обладающими большим приоритетом. При этом, если элемент обладает низким приоритетом, но не вызывает конфликтов с другими элементами, он может не исключаться из сети.

После завершения обработки семантической сети выполняется построение результирующего множества, содержащего те и только те элементы семантической сети, типы которых соответствуют семантике предметной области. Для этого выполняется фильтрация семантической сети, а маркеры и маркерные связи преобразуются в тона семантических объектов.

Модуль обработки текста является неотключаемым компонентом сервера.

3.1.1.4.5 Модуль постобработки

Модуль постобработки предназначен для объединения результирующих множеств (семантических сетей), полученных в результате отдельной обработки фрагментов, выделенных фрагментатором. К функциям модуля постобработки относятся:

- идентификация семантических объектов, принадлежащих результирующим множествам различных фрагментов, по совпадению значений атрибутов;
- вычисление значений хэш-функций семантических объектов в соответствии с определёнными для предметной области настройками хэширования объектов.

Вычисление значений хэш-функций для семантических объектов выполняется на завершающем этапе обработки данных и обеспечивает возможность идентификации этих объектов средствами прикладной информационной системы. Хэш-функции вычисляются на значениях ключевых наборов атрибутов, определяемых для типов семантических объектов в составе общих настроек предметной области.

При вычислении значений хэш-функций могут использоваться данные о лексической и синтаксической структуре исходных данных, представленных в формате частично структурированных документов (это позволяет, в частности, получать различные значения хэш-функций для объектов с совпадающими значениями ключевых наборов атрибутов, описания которых содержатся в различных предложениях, абзацах обрабатываемого текста или различных текстах).

Модуль постобработки является неотключаемым компонентом сервера. Настраиваемыми параметрами модуля являются настройки хэширования семантических объектов (см. п. 3.7.2.2).

3.1.1.5 Подсистема управления настройкой на предметную область

Подсистема управления настройкой на предметную область предназначена для хранения и управления данными, определяющими настройку ЛП на конкретную предметную область. В состав подсистемы входят:

- модуль управления конфигурацией предметной области;
- модуль управления словарями;
- средство разбора (парсер) правил предметной области;
- модуль подключения функций из внешних библиотек;
- модуль верификации конфигурации предметной области.

Настройки ЛП на предметную область определяются в конфигурационных файлах, располагающихся в отдельном каталоге файловой системы или в едином архиве формата

ZIP (в последнем случае обеспечивается защита конфигурационных файлов от несанкционированного изменения путем размещения соответствующих файлов подписей).

3.1.1.5.1 Модуль управления конфигурацией предметной области

Модуль управления конфигурацией предметной области предназначен для хранения перечня типов семантических объектов и связей предметной области, а также списка используемых этой предметной областью словарей, правил и др. К функциям модуля относятся:

- загрузка структуры конфигурационных файлов предметной области;
- загрузка перечня типов семантических объектов и связей из соответствующего конфигурационного файла;
- загрузка настроек хэширования семантических объектов из соответствующего конфигурационного файла;
- инициирование загрузки конфигурационных файлов другими компонентами подсистемы настройки на предметную область;
- предоставление данных о конфигурации предметной области по запросам других компонентов сервера.

Модуль управления конфигурацией предметной области является неотключаемым компонентом сервера. Параметры модуля и используемые им данные хранятся в конфигурационных файлах `config_part.xml` и `result_types.xml`, рассмотренные в п. 3.7.2.2.

3.1.1.5.2 Модуль управления словарями

Модуль управления словарями предназначен для хранения словарей предметной области и предоставления данных этих словарей другим компонентам сервера. К функциям модуля управления словарями относятся:

- загрузка словарей в соответствии с параметрами конфигурационных файлов;
- определение семантических доменов для элементов словарей по запросам компонентов сервера.

С технической точки зрения словари делятся на словари, сопоставляющие заданному значению единственный семантический домен (например, словари категорий, словари тонов) и словари, сопоставляющие заданному значению несколько семантических доменов (например, словари расшифровок, словари транслитераций). В зависимости от настройки на предметную область любой из словарей может быть интерпретирован с точки зрения его назначения различным образом (см. основные типы интерпретации словарей в п. 2.2.4.2.3).

Модуль подключения словарей является неотключаемым компонентом сервера. Настраиваемыми параметрами модуля являются перечень файлов, содержащих словари, и типы интерпретации этих словарей (см. п. 3.7.2.3).

3.1.1.5.3 Средство разбора (парсер) правил предметной области

Парсер правил предметной области предназначен для загрузки правил преобразования семантической сети (из файлов формата XML) и предоставления этих данных семантической подсистеме сервера. К функциям модуля относятся:

- загрузка правил преобразования семантической сети;
- контроль синтаксической корректности и семантической целостности правил преобразования семантической сети.

Парсер правил является неотключаемым компонентом сервера, настраиваемыми параметрами которого являются правила преобразования семантической сети. Описание правил преобразования семантической сети в п. 3.7.2.4.

3.1.1.5.4 Модуль подключения функций внешних библиотек

Модуль подключения функций внешних библиотек предназначен для обеспечения возможности использования правилами обработки семантической сети прикладных функций, размещенных во внешних библиотеках. К функциям модуля относятся:

- первоначальная загрузка всех внешних библиотек для обращения к их функциям;
- загрузка словарей, используемых функциями внешних библиотек, в соответствии с их конфигурационными файлами (манифестами);
- создание экземпляров функций и соответствующих интерфейсов для обращения к ним по запросам компонентов сервера;
- обработка ошибок и исключительных ситуаций, возникающих при выполнении функций внешних библиотек.

Внешние библиотеки представляют собой отдельные программные модули, предоставляющие (экспортирующие) точки входа для обращения к их подпрограммам (функциям). Внешние библиотеки представляют собой динамические библиотеки в форматах исполняемых файлов соответствующих операционных систем (Win32, Linux, Mac OS X).

Модуль подключения функций внешних библиотек является неотключаемым. Формат конфигурационного файла (манифеста) внешней библиотеки, определяющий перечень используемых этой библиотекой словарей, рассмотрен в п. 3.7.2.5.

3.1.1.5.5 Модуль верификации конфигурации предметной области

Модуль верификации конфигурации предметной области предназначен для автоматического контроля целостности и подлинности конфигурационных файлов предметной области в процессе инициализации сервера ЛП. К функциям модуля относятся:

- вычисление контрольной суммы конфигурационных файлов в соответствии с алгоритмом SHA-512;
- автоматический контроль соответствия вычисленной контрольной суммы значению, гарантирующему подлинность конфигурационных файлов.

Сервер ЛП обращается к функциям модуля верификации конфигурации предметной области в случае, когда конфигурационные файлы предметной области хранятся в виде архива формата ZIP. В случае, если результат верификации конфигурационных файлов оказывается отрицательным, выводится сообщение о повреждении конфигурационных файлов. Модуль верификации конфигурации предметной области является неотключаемым и не обладает настраиваемыми параметрами.

3.1.2 Агент-планировщик

Агент-планировщик предоставляет клиентским приложениям сетевой протокол для формирования и выполнения запросов к лингвистическому процессору. Клиентское приложение, формирующее запрос, указывает в соответствующих полях запроса идентификаторы предметных областей, в соответствии с настройками которых должны обрабатываться текстовые поля запроса (для плоского текста и DOM-документа используется одна предметная область для всего документа).

Идентификаторы предметных областей анализируются агентом-планировщиком, который либо передает запрос для обработки одному из запущенных ранее серверов, либо инициирует запуск нового сервера с необходимыми настройками предметной области (при этом в целях балансировки нагрузки могут запускаться несколько серверов с одинаковыми настройками предметной области).

Таким образом, для выполнения заданий по обработке текстов, относящихся к различным предметным областям, запускаются несколько экземпляров сервера, каждый со своей настройкой на предметную область. Запуск серверов, направление им запросов клиентов и остановку серверов выполняет агент-планировщик.

Сетевой протокол агента-планировщика содержит следующие группы методов:

- прикладные методы;
- диагностические методы.

Прикладные методы предназначены для обработки запросов клиентского приложения и обрабатывают следующие виды запросов:

- асинхронный запрос на обработку текста, частично структурированного документа или DOM-документа;
- асинхронный запрос на получение результатов выполнения запроса на обработку текста, частично структурированного документа или DOM-документа.

Диагностические методы предназначены для обработки запросов о конфигурации предметной области и состоянии обработки прикладных запросов и обрабатывают следующие виды запросов:

- асинхронный запрос информации о конфигурации предметной области;
- синхронный запрос на получение статуса любого асинхронного запроса;
- синхронный запрос на получение результатов выполнения запроса о конфигурации предметной области.

С технической точки зрения агент-планировщик представляет собой многопоточный процесс (служба на платформе Win32 или демон в UNIX-подобных операционных системах).

Сетевой протокол агента-планировщика реализован по технологии XML over HTTP. В качестве транспортного протокола используется HTTP, в теле сообщения которого размещается XML. Запросы осуществляются методом POST, допустимый статус HTTP-ответа – 200.

Методы сетевого протокола агента-планировщика реализуют обработку асинхронных запросов по модели polling. На каждый асинхронный запрос возвращается текстовый ключ (cookie), по которому в дальнейшем может быть получен статус или результат выполнения этого запроса.

Примеры использования методов сетевого протокола рассмотрены в п. 3.2.

3.2 Решения по взаимосвязям со смежными системами, обеспечению совместимости

Лингвистический процессор «АРИОН-Лингво» представляет собой серверное приложение, выполняющее запросы на обработку данных, поступающие от клиентских приложений. Таким образом, решения по взаимосвязям ЛП с внешними системами определяются форматами входных и выходных данных запросов, поступающих через клиентский сетевой протокол агента-планировщика.

3.2.1 Входные данные

Входные данные клиентских запросов к ЛП «АРИОН-Лингво» подразделяются на следующие виды:

- плоский текст (информационное сообщение);
- частично формализованные документы;
- DOM-документы.

3.2.1.1 Плоский текст

Входные данные в виде плоского текста представляют собой текстовые файлы, содержащие текст на русском или английском языках в произвольной кодировке без форматирования. Для корректной передачи текстового файла по протоколу HTTP плоский текст помещается в тэги `<decl>` и `<text>`; кроме того, при помощи дополнительного тэга `<route>` указывается идентификатор предметной области (параметр `knowledgeRoot`).

Пример входных данных в виде плоского текста:

```
<decl>
  <route knowledgeroot="ARION-FS" serverid="" />
  <text>
    ИноСМИ.Ru | Глава молдавского государства втянул
    Молдову в международный скандал ("Независимая
    Молдова", Молдавия)
    15 октября 2009
    Официальный представитель МИД России выступил с
    заявлением в связи с допущенной Михаем Гимпу
    "односторонней интерпретацией" его дискуссии с
    президентом России Дмитрием Медведевым.
    Заявление официального представителя МИД России
    Андрея Нестеренко в частности касались утверждений
    Гимпу относительно якобы высказанного им российскому
    президенту «требования вывести армию». "Первый
    контакт Медведева с Гимпу, который, будучи спикером
    парламента, временно исполняет президентские
    обязанности, на что-то большее, чем обзор отношений,
    претендовать не мог", - отметил Нестеренко.
    "Конечно, каждый политик вправе додумывать
    впечатления от общения с партнерами. Но выстраивать
    на собственной односторонней интерпретации
    конструкцию договоренностей - это нечто иное". -
```

сказал официальный представитель МИД. Российский дипломат допустил, что "формирование структур власти в Молдавии, уточнение политических концепций в рамках ее правящей коалиции продолжается не без противоречивых сигналов".

Встреча врио президента Михая Гимпу и президента Дмитрия Медведева имела место в Кишиневе в рамках Саммита глав государств СНГ в пятницу 9 октября. Журналисты были проинформированы о результатах и содержании встречи Михаем Гимпу на созванной им 10 октября пресс-конференции.

</text>

</decl>

Входные данные в виде плоского текста передаются агентом-планировщиком серверу обработки неструктурированной информации, настроенному на соответствующую предметную область, без изменений.

3.2.1.2 Частично структурированные документы

Частично структурированные документы представляют собой XML-файлы, содержащие как описания объектов предметной области, так и фрагменты плоского текста, ассоциированные с этими объектами. Формат частично структурированного документа включает следующие конструкции языка XML:

- корневая секция <arionDocument>, определяющая XSD-схемы и пространства имен XML-документа и содержащая основные секции <objects>, <links>, <syntaxTrees> и <sources>;
- секция <objects>, содержащая перечень семантических объектов предметной области (тэги <object>);
- тэг <object>, определяющий один семантический объект предметной области и задаваемый идентификатором (параметр id), техническими признаками (параметры material, archived), типом (тэг <type>), набором атрибутов (тэг <attributes>), набором дополнительных характеристик (тэг <tags>) и набором лексических ссылок (тэг <coreference>). Если для некоторого объекта material="true", то с этим объектом ассоциировано полнотекстовое поле;
- секция <attributes> семантического объекта, определяющая набор атрибутов объекта и их значений (тэги <attribute>);
- тэг <attribute>, определяющий значение одного атрибута семантического объекта и задаваемый текстовым названием (тэг <name>) и текстовым значением (тэг <value>);
- секция <tags> семантического объекта, определяющая дополнительные характеристики этого объекта (тэги <tag>);
- тэг <tag>, определяющий одну дополнительную характеристику семантического объекта и задаваемый типом (параметр role) и внутренними атрибутами. Состав внутренних атрибутов определяется значением параметра role, например, для значения role="MARKER" набор атрибутов включает

- текстовое название маркера (тэг <name>), его текстовое значение (тэг <value>) и вес (тэг <weight>);
- секция <coreference> семантического объекта, определяющая лексические ссылки (тэги <anaphor>) на этот объект;
 - тэг <anaphor>, означающий, что текст, соответствующий другому объекту документа, определяемому параметром id, представляет собой описание текущего семантического объекта;
 - секция <links>, определяющая семантические связи (тэги <link>) между описанными ранее объектами;
 - тэг <link>, определяющий один экземпляр семантической связи и задаваемый идентификатором (параметр id), техническими признаками (параметр archived), типом (тэг <type>), идентификаторами связываемых объектов (параметры object1, object2), значением (тэг <verb>) и окраской (тэг <linkColor>);
 - секция <syntaxTrees>, определяющая синтаксические структуры (тэги <location>), выделенные в полнотекстовых полях, ассоциированных с семантическими объектами;
 - тэг <location>, имеющий одно из трёх значений:
 - если параметр type имеет значение "text", то тэг задает корень синтаксического дерева полнотекстового поля, ассоциированного с объектом, идентификатор которого указан в параметре materialref;
 - если параметр type имеет значение "object", то тэг определяет один из экстенгов (отрезков текста), соответствующих объекту, идентификатор которого указан в параметре objref. При этом для экстенга указывается родительская синтаксическая структура (параметр parent), смещение экстенга в полнотекстовом поле (параметр start), длина экстенга в символах (параметр length);
 - если параметр type имеет значение "group" или "clause", то тэг описывает синтаксическую структуру, входящую в синтаксическое дерево некоторого полнотекстового поля. Для синтаксической структуры указываются родительская структура (параметр parent), смещение синтаксической структуры в полнотекстовом поле (параметр start), ее длина в символах (параметр length), а также (опционально) тип соответствующей синтаксической группы или клаузы (тэг <description>).
 - секция <sources>, описывающая неструктурированные полнотекстовые поля (тэги <source>), подлежащие обработке лингвистическим процессором;
 - тэг <source>, описывающий одно полнотекстовое поле и задаваемый идентификатором объекта, с которым ассоциировано поле (параметр materialref), признаком необходимости обработки этого поля средствами ЛП (параметр lingvo), идентификатором предметной области (параметр knowledgeRoot), значением поля (тэг <text>) и ссылкой на первоисточник (тэг original).

Пример входных данных ЛП в виде частично структурированного документа:

```
<?xml version="1.0" encoding="UTF-8" ?>
<decl>
<route knowledgeroot="" serverid=""/>
<arionDocument xmlns="http://schemas.sytech.ru/arionDocument"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://schemas.sytech.ru/arionDocumen
t arionDocument.xsd" version="1.0" namespace="decl">
  <objects>
    <object id="1" material="true" archived="false">
      <type>foobar</type>
    </object>
    <object id="2" material="true" archived="false">
      <type>foobar</type>
    </object>
    <object id="100" archived="false">
      <type>СМИ</type>
      <attributes>
        <attribute>
          <name>НАЗВАНИЕ</name>
          <value>ИЗВЕСТИЯ</value>
        </attribute>
      </attributes>
      <tags>
        <tag role="MARKER">
          <type>ДОСТОВЕРНОСТЬ</type>
          <value>ВЫСОКАЯ</value>
          <weight>100</weight>
        </tag>
      </tags>
      <coreference>
        <anaphor id="200" />
      </coreference>
    </object>
    <object id="200" archived="false">
      <type>ИСТОЧНИК</type>
    </object>
  </objects>
  <links>
    <link id="800" archived="false" object1="100"
      object2="200">
      <type>СОВПАДЕНИЕ</type>
      <verb>АНАЛОГИЧНЫЙ_ИСТОЧНИК</verb>
      <linkColor
        direction="SYMMETRICAL">baz</linkColor>
    </link>
    <link id="900" archived="false" object1="100"
```

```

    object2="200">
      <type>ПОДТВЕРЖДЕНИЕ</type>
      <verb>СООТВЕТСТВИЕ_СВЕДЕНИЙ</verb>
      <linkColor
        direction="SYMMETRICAL">baz</linkColor>
    </link>
  </links>
  <syntaxTrees>
    <location id="11" type="text" materialref="1" />
    <location id="12" type="text" materialref="2" />
    <location id="20" type="clause" start="2" length="3"
      parent="11" />
    <location id="30" type="group" start="2"
      length="33"/>
    <location id="50" type="group" start="2"
      length="33">
      <description>foobar</description>
    </location>
    <location id="60" type="object" start="222"
      length="333" parent="11" objref="200" />
    <location id="70" type="object" start="555"
      length="777" parent="12" objref="200" />
  </syntaxTrees>
  <sources>
    <source materialref="2" lingvo="true"
      knowledgeRoot="ARION-FS">
      <text>Депутаты фракций ЛДПР и "Справедливая
        Россия", покинувшие заседание Госдумы 14
        октября, 16 октября прекратили акцию протеста и
        вернулись в парламент. Таким образом, на
        продолжении акции до встречи с президентом РФ
        Дмитрием Медведевым до сих пор продолжает
        настаивать лишь КПРФ.</text>
      <original fs2File="fs2://foobar.pdf" />
    </source>
  </sources>
</arionDocument>
</decl>

```

Формат частично структурированного документа используется также для структурированного описания информационных объектов, являющегося результатом работы лингвистического процессора (см. п. 3.2.2.1).

3.2.1.3 DOM-документы

DOM-документы представляют собой XML-файлы, определяющие особенности форматирования и расположения фрагментов текста HTML-документа. Формат DOM-документа включает следующие конструкции языка XML:

- корневая секция <dom>, соответствующая полному документу, задаваемая названием документа (параметр title) и ссылкой на его источник (параметр url), а также содержащая элементы дерева DOM (тэги <element>);
- секция <element>, описывающая структурный элемент DOM-документа и содержащая фрагмент текста документа (тэг <text>), параметры форматирования (тэг <styles>), атрибуты соответствующего HTML-тэга (тэг <attributes>), а также другие элементы (тэги <element>). Структурный элемент DOM-документа описывается идентификатором (параметр id), текстовым названием (параметр name), координатами элемента в документе (параметры left, top), размерами элемента (параметры width, height), размером шрифта для текста, соответствующего элементу (параметр font-size);
- тэг <text>, содержащий фрагмент плоского текста документа, соответствующий элементу (может отсутствовать);
- секция <styles>, определяющая параметры форматирования текста элемента и содержащая пары «параметр–значение» (тэги <property>);
- секция <attributes>, содержащая значения атрибутов HTML-тэгов, соответствующих элементам DOM-документа (тэги <attr>);
- тэг <property>, описывающий один параметр форматирования фрагмента текста и задаваемый текстовым названием параметра (параметр name) и текстовым значением (внутренний текст тэга <property>);
- тэг <attr>, описывающий один атрибут тэга языка HTML, соответствующего элементу и задаваемый текстовым названием атрибута (параметр name) и текстовым значением (внутренний текст тэга <attr>).

Фрагмент входного файла ЛП, соответствующего плоскому тексту примера, приведенного в п. 3.2.1.1, в виде DOM-документа:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<dom title="ИноСМИ.Ru | Глава молдавского государства втянул
Молдову в международный скандал">
  <element font-size="12" height="1302" id="0" left="207"
    name="TD" top="241" width="569">
    <styles>
      <property name="font-family">Times New
        Roman</property>
      <property name="font-size">12pt</property>
      <property name="font-style">normal</property>
      <property name="font-variant">normal</property>
      <property name="font-weight">400</property>
      <property name="border-width">medium</property>
      <property name="border-style">none</property>
```

```
<property name="border-color">
  #d4d0c8</property>
<property name="background-color">
  transparent</property>
<property name="color">#000000</property>
<property name="line-height">normal</property>
<property name="text-align">left</property>
<property name="text-indent">0pt</property>
<property name="text-transform">none</property>
<property name="text-decoration">
  none</property>
</styles>
<element font-size="12" height="177" id="1"
  left="207" name="TABLE" top="274" width="563">
  <attributes>
    <attr name="border">0</attr>
    <attr name="cellSpacing">0</attr>
    <attr name="cellPadding">0</attr>
    <attr name="width">563</attr>
  </attributes>
  <styles>
    <property name="font-family">Times New
      Roman</property>
    <property name="font-size">12pt</property>
    ...
  </styles>
  <element font-size="12" height="177" id="2"
    left="207" name="TBODY" top="274"
    width="563">
    <attributes>
      <attr name="vAlign">top</attr>
      <attr name="width">100%</attr>
    </attributes>
    <styles>
      <property name="font-family">Times
        New Roman</property>
      ...
    </styles>
    ...
    <element id="7" name="text">
      <text>Глава молдавского государства
        втянул Молдову в международный
        скандал</text>
    </element>
  </element>
  <element id="8" name="text">
    <text>("Независимая Молдова",
      Молдавия)</text>
```

```
</element>
<element id="9" name="text">
  <text>15 октября 2009</text>
</element>
...
</element>
...
</element>
<element font-size="12" height="21" id="12" left="207"
name="TABLE" top="451" width="563">
<attributes>
  <attr name="border">0</attr>
  <attr name="cellSpacing">0</attr>
  <attr name="cellPadding">0</attr>
  <attr name="width">563</attr>
</attributes>
...
<element font-size="6.0492" height="14" id="16"
left="207" name="A" top="458" width="100">
<attributes>
  <attr name="href">mailto:?Subject=
Публикации ИноСМИ.Ru&Body=Глава молдавского
государства втянул Молдову в международный
скандал http://www.inosmi.ru/translation/
253628.html</attr>
</attributes>
<styles>
  <property name="font-family">"Arial Cyr",
Arial, Helvetica, sans-serif</property>
...
</styles>
<element id="18" name="text">
  <text>[отослать ссылку]</text>
</element>
</element>
<element id="29" name="text">
  <text>Официальный представитель МИД России
выступил с заявлением в связи с допущенной
Михаеом Гимпу "односторонней интерпретацией"
его дискуссии с президентом России</text>
</element>
<element font-size="12" height="19" id="30"
left="352" name="A" top="554" width="163">
<attributes>
  <attr name="style" />
```

```
<attr name="title">Искать в архиве:
"Дмитрий Медведев"</attr>
<attr name="href">http://www.inosmi.ru/
earch/?query=%22%С4%EC%F2%F0%E8%E9%20%CC
%E5%E4%E2%E5%E4%E5%E2%22&from_hs=1</attr>
</attributes>
<styles>
  <property name="font-family">"Times New
  Roman Cyr", "Times New Roman"</property>
  <property name="font-size">12pt</property>
  ...
</styles>
<element id="31" name="text">
  <text>Дмитрием Медведевым</text>
</element>
</element>
<element id="32" name="text">
  <text>.</text>
</element>
<element id="33" name="text">
  <text>Заявление официального представителя МИД
  России Андрея Нестеренко в частности касались
  утверждений Гимпу относительно якобы
  высказанного им российскому президенту
  "требования вывести армию".</text>
</element>
<element id="34" name="text">
  <text>"Первый контакт Медведева с Гимпу,
  который, будучи спикером парламента, временно
  исполняет президентские обязанности, на что-то
  большее, чем обзор отношений, претендовать не
  мог", - отметил Нестеренко.</text>
</element>
...
</element>
<element id="59" name="text">
  <text>Саммит СНГ в европейском формате</text>
</element>
<element id="60" name="text">
  <text>("MoldovaNova", Молдавия)</text>
</element>
<element font-size="12" height="19" id="61" left="207"
  name="A" top="1485" width="403">
  <attributes>
    <attr name="href">http://www.inosmi.ru/
    translation/53410.html</attr>
    ...
```

```
</attributes>
<styles>
  <property name="font-family">"Times New Roman
    Cyr", "Times New Roman"</property>
  ...
</styles>
<element id="62" name="text">
  <text>Молдавская демократия нуждается в
    поддержке Евросоюза</text>
</element>
</element>
<element id="63" name="text">
  <text>("EUobserver.com", Бельгия)</text>
</element>
</element>
</dom>
```

Входные данные в формате DOM-документа формируются на основе HTML-документа при помощи специализированного плагина для браузера Internet Explorer или средства для пакетного преобразования HTML-документов к DOM-документам (специализированный прокси-сервер).

3.2.2 Выходные данные

Выходными данными (результатом работы) ЛП «АРИОН-Лингво» является результирующее множество (структурированные описания семантических объектов, значений их атрибутов и связей между объектами).

3.2.2.1 Результирующее множество

Результирующее множество является основным результатом работы лингвистического процессора и представляет собой набор структурированных описаний семантических объектов, полученных в результате обработки входных данных, в формате XML.

Результирующее множество, в зависимости от вида входных данных, может быть представлено в двух форматах:

- базовый формат результирующего множества;
- формат аналитического документа.

3.2.2.1.1 Базовый формат результирующего множества

Базовый формат результирующего множества используется для представления результатов обработки ЛП входных данных, представленных в виде плоского текста или DOM-документа и представляет собой XML-файл, содержащий описания семантических объектов предметной области, значения их атрибутов и связи между объектами. Базовый формат результирующего множества включает следующие конструкции языка XML:

- корневая секция <decl>, определяющая версию формата XML-документа и содержащая основные секции <objects>, <links> и <metadata>;
- секция <objects>, содержащая перечень семантических объектов предметной области (тэги <object>);

- тэг <object>, определяющий один семантический объект предметной области и задаваемый идентификатором (параметр id), уровнем значимости (параметр significance), типом (тэг <type>), значением (тэг <value>), расположением в тексте (тэг <extents>), набором атрибутов (тэг <attributes>) и набором тонов (тэг <tones>);
- секция <extents> семантического объекта, определяющая набор фрагментов лексической структуры, соответствующих объекту (тэги <extent>);
- тэг <extent>, определяющий один фрагмент лексической структуры, соответствующий семантическому объекту, и задаваемый ссылкой на синтаксическую структуру (параметр relation), индексом первой лексемы фрагмента (параметр start) и длиной фрагмента (параметр len);
- секция <attributes> семантического объекта, определяющая набор атрибутов объекта и их значений (тэги <attribute>);
- тэг <attribute>, определяющий значение одного атрибута семантического объекта и задаваемый текстовым названием (тэг <name>) и текстовым значением (тэг <value>);
- секция <tones> семантического объекта, определяющая тональные характеристики этого объекта (тэги <tone>);
- тэг <tone>, определяющий одну тональную характеристику семантического объекта и задаваемый весом (параметр wight), типом характеристики (тэг <type>) и её значением (тэг <value>);
- секция <links>, определяющая семантические связи (тэги <link>) между описанными ранее объектами;
- тэг <link>, определяющий один экземпляр семантической связи и задаваемый собственным идентификатором (параметр id), идентификаторами связываемых объектов (параметры object1, object2), окраской (тэг <verb>), типом (тэг <type>) и приоритетом (тэг <prior>);
- секция <metadata>, определяющая синтаксические структуры (тэги <syntaxLocations>), выделенные в тексте;
- тэг <syntaxLocation>, описывающий одну синтаксическую структуру и задаваемый идентификатором (параметр id), индексом первой лексемы (параметр start), индексом последней лексемы (параметр last), типом (параметр locationType) и идентификатором родительской синтаксической структуры (параметр relations).
- секция <coreferences>, описывающая ссылочные связи между лексемами (тэги <coreference>);
- тэг <coreference>, описывающий одну ссылочную связь между лексемами, задаваемую идентификатором (параметр id) и значением ссылки (тэг <anaphor>);
- тэг <anaphor>, описывающий одну анафорическую связь между лексемами с идентификатором id.

Пример результирующего множества в базовом формате:

```
<?xml version="1.0" encoding="UTF-8"?>
<decl version="2.6">
  <objects>
    <object id="1" significance="1">
      <type>ГОСУДАРСТВО</type>
      <value>АФГАНИСТАН</value>
      <extents>
        <extent relation="16" start="711" len="10"/>
      </extents>
      <attributes>
        <attribute>
          <name>ОФИЦИАЛЬНОЕ_НАИМЕНОВАНИЕ</name>
          <value>АФГАНИСТАН</value>
        </attribute>
      </attributes>
      <tones>
        <tone weight="2">
          <type>ТОНАЛЬНОСТЬ</type>
          <value>ПОМОЩЬ</value>
        </tone>
      </tones>
    </object>
    <object id="2" significance="1">
      <type>ДАТА</type>
      <value>11.03.2004</value>
      <extents>
        <extent relation="12" start="183" len="15"/>
      </extents>
      <attributes>
        <attribute>
          <name>ГОД</name>
          <value>2004</value>
        </attribute>
        <attribute>
          <name>ДЕНЬ</name>
          <value>11</value>
        </attribute>
        <attribute>
          <name>МЕСЯЦ</name>
          <value>03</value>
        </attribute>
      </attributes>
    </object>
  </objects>
</decl>
```

```
<tones/>
</object>
</objects>
<links>
  <link id="3" object1="2" object2="12">
    <verb>ЗАЩИЩАТЬ</verb>
    <type>ГЕОГРАФИЧЕСКАЯ_ПРИВЯЗКА</type>
    <prior>124</prior>
  </link>
  <link id="5" object1="21" object2="10">
    <verb>В</verb>
    <type>МЕСТО_ДЕЙСТВИЯ</type>
    <prior>96</prior>
  </link>
  <link id="6" object1="20" object2="13">
    <verb>ГОРОД МАДРИД</verb>
    <type>СВЯЗЬ_ПРОИСШЕСТВИЕ_ДАТА</type>
    <prior>128</prior>
  </link>
</links>
<metadata>
  <syntaxLocations>
    <syntaxLocation id="7" start="0" last="849"
      locationType="text">
      <type/>
    </syntaxLocation>
    <syntaxLocation id="8" start="0" last="115"
      locationType="sentence" relations="7">
      <type/>
    </syntaxLocation>
    <syntaxLocation id="9" start="0" last="114"
      locationType="clause" relations="7">
      <type>КР_ПРЧ</type>
    </syntaxLocation>
    <syntaxLocation id="10" start="116" last="202"
      locationType="clause" relations="8">
      <type>ГЛ_ЛИЧН</type>
    </syntaxLocation>
    <syntaxLocation id="11" start="204" last="240"
      locationType="clause" relations="8">
      <type>ГЛ_ЛИЧН</type>
    </syntaxLocation>
    <syntaxLocation id="12" start="277" last="290"
      locationType="clause" relations="8">
      <type>ЧАСТЬ</type>
    </syntaxLocation>
  </syntaxLocations>
</metadata>
</document>
```

```

<syntaxLocation id="13" start="11" last="15"
locationType="group" relations="9">
  <type>ПГ</type>
</syntaxLocation>

<syntaxLocation id="14" start="97" last="113"
locationType="group" relations="9">
  <type>ЧИСЛ_СУЩ</type>
</syntaxLocation>

<syntaxLocation id="15" start="819" last="833"
locationType="group" relations="10">
  <type>ПРИЛ_СУЩ</type>
</syntaxLocation>

<syntaxLocation id="16" start="836" last="847"
locationType="group" relations="11">
  <type>ПРЯМ_ДОП</type>
</syntaxLocation>

</syntaxLocations>
<coreferences>
  <coreference id="17">
    <anaphor id="17"/>
  </coreference>
  <coreference id="18">
    <anaphor id="18"/>
  </coreference>
</coreferences>
</metadata>
</decl>

```

3.2.2.1.2 Формат частично структурированного документа

Формат частично структурированного документа применяется для представления результатов обработки ЛП входных данных, также представленных в виде частично структурированного документа. Формат результирующей семантической сети полностью аналогичен формату входных данных в виде частично структурированных документов (см. п. 3.2.1.2).

Ниже приведен фрагмент результирующего множества в виде частично структурированного документа, полученного в результате применения ЛП к примеру входных данных, рассмотренному в п. 3.2.1.2.

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<decl>
  <arionDocument
    xmlns="http://schemas.sytech.ru/arionDocument"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" namespace="decl" version="1.0"
    xsi:schemaLocation="http://schemas.sytech.ru/arionDo

```

```
cument arionDocument.xsd">
<objects>
  <object id="2000" archived="false">
    <type>ДАТА</type>
    <value>2003 ГОД</value>
    <attributes>
      <attribute>
        <name>ГОД</name>
        <value>2003</value>
      </attribute>
    </attributes>
    <extents />
  </object>
  <object id="2020" archived="false">
    <type>ФИЗ_ЛИЦО</type>
    <value>ПАУЭЛЛ КОЛИН</value>
    <attributes>
      <attribute>
        <name>ИМЯ</name>
        <value>КОЛИН</value>
      </attribute>
      <attribute>
        <name>ОТЧЕСТВО</name>
        <value />
      </attribute>
      <attribute>
        <name>ФАМИЛИЯ</name>
        <value>ПАУЭЛЛ</value>
      </attribute>
    </attributes>
    <extents />
  </object>
  <object id="2140" archived="false">
    <type>СТРАНА</type>
    <value>РОССИЯ</value>
    <attributes>
      <attribute>
        <name>НАЗВАНИЕ</name>
        <value>РОССИЯ</value>
      </attribute>
    </attributes>
    <extents />
  </object>
  <object id="2240" archived="false">
    <type>ПРОИСШЕСТВИЕ</type>
    <value>ОПЕРАЦИЯ</value>
    <attributes>
      <attribute>
        <name>ТИП_СОБЫТИЯ</name>
        <value>ОПЕРАЦИЯ</value>
```

```
        </attribute>
      </attributes>
      <extents />
    </object>
    ...
  </objects>
  <links>
    <link archived="false" id="800" object1="100"
      object2="200">
      <type>foo</type>
      <verb>bar</verb>
      <linkColor
        direction="SYMMETRICAL">baz</linkColor>
    </link>
    ...
  </links>
  <syntaxTrees>
    <location id="11" materialref="1"
      type="text" />
    <location id="1000" type="text"
      materialref="1" />
    <location id="2001" type="object" parent="1000"
      objref="2000" start="178" length="9" />
    <location id="2011" type="object" parent="1000"
      objref="2010" start="24" length="8" />
    <location id="2061" type="object" parent="1000"
      objref="2060" start="457" length="20" />
    <location id="2141" type="object" parent="1000"
      objref="2140" start="240" length="6" />
    <location id="2181" type="object" parent="1000"
      objref="2180" start="226" length="11" />
    <location id="2241" type="object" parent="1000"
      objref="2240" start="386" length="8" />
    <location id="3000" parent="1000"
      type="paragraph" start="0" last="725" />
    <location id="3001" parent="3000"
      type="sentence" start="0" last="125" />
    <location id="3006" parent="3001" type="clause"
      start="0" last="72">
      <description>ГЛ_ЛИЧН</description>
    </location>
    ...
    <location id="3022" parent="3021" type="clause"
      start="625" last="630">
```

```

        <description>ЧАСТЬ</description>
    </location>
    <location id="3023" parent="3006" type="group"
        start="10" last="22">
        <description>НАР_ЧИСЛ_СУЩ</description>
    </location>
    <location id="3028" parent="3010" type="group"
        start="134" last="149">
        <description>ГЕНИТ_ИГ</description>
    </location>
    <location id="3029" parent="3010" type="group"
        start="151" last="162">
        <description>ПРИЛ_СУЩ</description>
    </location>
    <location id="3041" parent="3012" type="group"
        start="343" last="354">
        <description>ОБОРОТ</description>
    </location>
</syntaxTrees>
<sources>
    <source lingvo="false" materialref="2">
        <text>bazbaz</text>
        <original fs2File="fs2://foobar.pdf" />
    </source>
    <source knowledgeRoot="ARION-FS" lingvo="true"
        materialref="2">
        <text>Депутаты фракций ЛДПР и "Справедливая
        Россия", покинувшие заседание Госдумы 14
        октября, 16 октября прекратили акцию
        протеста и вернулись в парламент. Таким
        образом, на продолжении акции до встречи с
        президентом РФ Дмитрием Медведевым до сих
        пор продолжает настаивать лишь КПРФ.</text>
        <original fs2File="fs2://foobar.pdf" />
    </source>
</sources>
</arionDocument>
</decl>

```

Результирующее множество, являющееся результатом работы ЛП «АРИОН-Лингво» и представленное в форме частично структурированного документа, может быть использовано в качестве входных данных для аналогичного ЛП. Такой подход предназначен для случаев, когда необходимо выполнить обработку документа на основе последовательного применения правил различных предметных областей.

3.3 Решения по режимам функционирования, диагностированию работы системы

ЛП «АРИОН-Лингво» рассчитан на функционирование в следующих режимах:

- рабочий режим;
- режим конфигурирования;
- режим диагностирования.

3.3.1 Рабочий режим

Рабочий режим является основным режимом функционирования ЛП. В этом режиме осуществляется выполнение клиентских запросов по обработке данных и диагностических запросов.

Переход ЛП в рабочий режим осуществляется автоматически при запуске программного обеспечения агента-планировщика. Запуск и останов серверов обработки неструктурированной информации осуществляется агентом-планировщиком по мере необходимости (см. п.п. 3.1.1, 3.1.1.5.5). В процессе запуска сервера ЛП проходят этап инициализации программного обеспечения (см. п. 3.3.2).

3.3.2 Режим конфигурирования

Режим конфигурирования является дополнительным режимом функционирования ЛП и включает два этапа:

- этап редактирования конфигурационных файлов;
- этап инициализации программного обеспечения ЛП.

На этапе редактирования конфигурационных файлов специалисты, обеспечивающие функционирование ЛП, могут изменять настройки программного обеспечения (см. п. 2.2.4.1) или конкретной предметной области (см. п. 2.2.4.2).

На этапе инициализации программного обеспечения функциональные компоненты считывают и анализируют соответствующие конфигурационные файлы и формируют внутренние структуры данных, соответствующие установленным специалистами параметрам. Этап инициализации программного обеспечения автоматически выполняется в полном объеме при запуске ЛП или может быть инициирован программно для отдельных предметных областей в рамках рабочего режима функционирования ЛП. На этапе инициализации программного обеспечения может выполняться верификация конфигурации предметной области (см. п. 3.1.1.5.5).

3.3.3 Режим диагностирования

Режим диагностирования является дополнительным режимом функционирования ЛП и заключается в анализе результатов функционирования программного обеспечения ЛП в целях оптимизации его характеристик или выявления и устранения ошибок конфигурации.

Основным способом анализа результатов функционирования ЛП является изучение протоколов обработки входных данных (см. п. 2.2.3.2) и выявление классов входных данных, обрабатываемых ЛП некорректно. Помимо этого, для диагностирования технических характеристик ЛП возможно использование диагностических методов клиентского сетевого протокола агента-планировщика (см. п. 3.1.1.5.5).

3.4 Решения по численности, квалификации и функциям персонала, режимам его работы, порядку взаимодействия

Для обеспечения функционирования ЛПП «АРИОН-Лингво» необходимо наличие в штате эксплуатирующего ее подразделения специалистов следующих категорий:

- технический администратор;
- инженер по знаниям;
- лингвист.

3.4.1 Технический администратор

К функциям технического администраторов ЛПП относятся установка и настройка системного программного обеспечения; установка и настройка ЛПП (в том числе – редактирование конфигурационных файлов сервера и агента-планировщика); проверка работоспособности технического и программного обеспечения; диагностика и устранение неполадок; выполнение регламентных операций (резервное копирование и восстановление данных).

Режим работы технического администратора ЛПП должен соответствовать регламенту рабочего режима функционирования программного обеспечения лингвистического процессора. В случае, если рабочий режим подразумевает круглосуточное функционирование ЛПП, в штатном расписании должна быть предусмотрена должность дежурного технического администратора для круглосуточного обеспечения нормального функционирования ЛПП.

При выявлении в работе ЛПП неполадок, являющихся следствием ошибок в конфигурации конкретной предметной области, технический администратор должен обратиться к инженеру по знаниям для диагностики и устранения обнаруженных ошибок.

3.4.2 Инженер по знаниям

К функциям инженера по знаниям относятся подготовка совокупности конфигурационных файлов, определяющих настройку лингвистического процессора на конкретную предметную область; контроль качества настройки на предметную область; разработка и отладка правил преобразования семантической сети для конкретных предметных областей; постановка лингвистам задач, связанных с наполнением и актуализацией используемых словарей; выявление и устранение ошибок в конфигурации предметной области.

Режим работы инженера по знаниям должен соответствовать регламенту работы подразделения, обеспечивающего функционирование автоматизированной информационной системы, в составе которой используется ЛПП.

При выявлении в работе ЛПП неполадок, являющихся следствием ошибочной настройки системного или прикладного программного обеспечения, инженер по знаниям должен обратиться к техническим администраторам для устранения обнаруженных ошибок. При необходимости разработки дополнительных или расширения (актуализации) существующих словарей предметной области (в том числе – в целях устранения ошибок в конфигурации предметной области) инженер по знаниям должен сформулировать соответствующую задачу лингвистам.

3.4.3 Лингвист

К функциям лингвиста относится наполнение и актуализация словарей конкретной предметной области; выявление и устранение ошибок в работе ЛП, являющихся следствием ошибок при подготовке словарей.

Режим работы лингвиста должен соответствовать регламенту работы подразделения, обеспечивающего функционирование автоматизированной информационной системы, в составе которой используется ЛП.

При выявлении в работе ЛП неполадок, которые не могут быть устранены в результате коррекции используемых словарей предметной области, лингвист должен обратиться к инженеру по знаниям или техническому администратору для устранения обнаруженных ошибок. Лингвист также может обращаться к инженеру по знаниям с инициативами по разработке дополнительных или расширения (актуализации) существующих словарей предметной области.

3.5 Сведения об обеспечении показателей качества

Основной потребительской характеристикой, определяющей качество лингвистического процессора, является возможность его применения для преобразования в структурированную форму текстовых данных, соответствующих различным предметным областям. В зависимости от особенностей предметной области лингвистический процессор должен различным образом интерпретировать слова и высказывания (последовательности слов, фрагмента DOM-документа) естественного языкового (или слабоструктурированного в случае DOM-документов) текста и формировать структуры данных, соответствующие семантике этих слов и высказываний с точки зрения этой предметной области.

ЛП «АРИОН-Лингво» реализует возможность обработки естественных языковых данных, соответствующих различным предметным областям, за счет использования двухуровневой схемы конфигурирования программного обеспечения ЛП. Такой подход позволяет настраивать механизмы обработки неструктурированной информации не только применительно к предметной области в целом, но и применительно к её отдельным секторам, что существенно повышает качество обработки данных.

Настройка ЛП «АРИОН-Лингво» в соответствии с особенностями конкретной предметной области осуществляется за счет редактирования соответствующих конфигурационных файлов (см. п.п. 3.7.1, 3.7.2). На первом уровне настройки определяются общие параметры функционирования ЛП при обработке данных в рамках этой предметной области:

- технические параметры асинхронного взаимодействия клиентского приложения, осуществляющего выполнение прикладных функций, с агентом-планировщиком и сервером обработки неструктурированной информации;
- общие параметры настройки на конкретные предметные области (идентификаторы предметных областей; языки и подмножества языков, используемые в рамках различных предметных областей, используемые кодировки; особенности структуры текстовых документов);
- технические параметры, определяющие особенности обработки входных данных за счет последовательного или параллельного использования нескольких экземпляров сервера обработки неструктурированной информации в целях повышения производительности ЛП;
- особенности применения серверов обработки неструктурированной информации, настроенных в соответствии с особенностями отдельных секторов предметной области, для обработки полнотекстовых полей частично структурированных документов, соответствующих этим секторам, что существенно повышает качество обработки входных данных;
- особенности использования в качестве входных данных текстовых документов, соответствующих любому из трех наиболее распространенных типов: плоский текст, гипертекст, структурированный документ с полнотекстовыми полями вне зависимости от технических особенностей их представления.

На втором уровне настройки определяются особенности обработки входных данных в контексте конкретной предметной области или её отдельного сектора:

- общая логика действий по обработке входных данных, которая задаётся многоуровневой системой правил преобразования семантической сети, соответствующей предметной области;

- перечень типов семантических объектов и связей, а также их атрибутов и характеристик, являющихся допустимыми в рамках выбранной предметной области;
- логика устранения неоднозначности входных данных, а также промежуточных и окончательных результатов их обработки – в рамках системы правил преобразования семантической сети;
- логика устранения опечаток, ошибок оператора, технических ошибок в кодировке, формате и др. – в рамках системы правил преобразования семантической сети и специализированных словарей;
- параметры ведения протокола обработки входных данных, позволяющего проанализировать процесс и результаты обработки конкретных текстов и актуализировать в соответствии с ними настройки ЛП на предметную область;
- централизованно хранимые и используемые словари категорий, расшифровок, тонов, транслитераций, опечаток, позволяющие корректно изменять данные о таксономии предметной области, не нарушая целостности данных о ней;
- дополнительные функции проверки условий и преобразования данных, обладающие произвольной внутренней логикой и имеющие доступ к словарям, размещаемые во внешних библиотеках и используемые правилами преобразования семантической сети.

Первоначальную настройку ЛП в соответствии с особенностями предметной области осуществляет разработчик. В дальнейшем настройка может быть актуализирована инженером по знаниям и лингвистом в рамках их функциональных обязанностей (см. п. 3.4).

Важной потребительской характеристикой ЛП, влияющей на его качество, является надёжность и стабильность функционирования. ЛП «АРИОН-Лингво» обладает рядом специализированных механизмов, обеспечивающих надёжное функционирование программного обеспечения и своевременное восстановление его после сбоев.

При возникновении сбоев в техническом или системном программном обеспечении, включая аварийное отключение электропитания, ЛП «АРИОН-Лингво» восстанавливает свою работоспособность после устранения сбоев и корректного перезапуска (за исключением случаев повреждения рабочих носителей информации с исполняемым программным кодом или конфигурационными файлами).

ЛП обеспечивает корректную обработку аварийных ситуаций, вызванных ошибочными действиями обслуживающего персонала, неверным форматом или недопустимыми значениями входных данных. В указанных случаях выводятся соответствующие аварийные сообщения, после чего программное обеспечение возвращается в рабочее состояние, предшествовавшее неверной (недопустимой) команде или некорректному вводу данных.

3.6 Состав функций, комплексов задач, реализуемых ЛП

ЛП реализует следующие комплексы задач:

- структурирование естественных языковых текстов (основная);
- генерация наборов правил преобразования семантической сети в соответствии с шаблонами, задаваемыми в терминах условий на значения атрибутов объектов и взаимосвязей предметной области;
- линеаризация (преобразование естественных языковых конструкций к форме, пригодной для дальнейшего использования в автоматизированных системах).

3.6.1 Структурирование естественных языковых текстов

Преобразование полнотекстовых полей входных данных к структурированному виду, определяемому общей моделью представления знаний о предметной области (см. п. 2.2.1) и конфигурацией конкретной предметной области (см. п. 2.2.4.2), является основной функцией ЛП «АРИОН-Лингво». В основу алгоритма структурирования естественных языковых текстов положено последовательное преобразование входных данных (естественноязыкового текста, частично структурированного естественноязыкового текста, DOM-документа) в результирующее множество, соответствующее информационной модели предметной области.

Последовательность этапов обработки информации при решении комплекса задач по структурированию естественных языковых текстов следующая:

- если исходные данные представлены в виде DOM-документа, то такой документ преобразуется в плоский текст, при этом формируется соответствующее отображение текста на DOM (см. п. 3.1.1.4.1);
- если исходные данные представлены в виде частично структурированного документа, то текст каждого полнотекстового поля этого документа извлекается для независимой обработки;
- на этапе препроцессирования естественной языковой текст приводится к единой кодировке; из текста удаляются лишние пробелы, переносы слов и специальные символы; в тексте выполняется замена символов для устранения простейших опечаток (см. п. 3.1.1.2);
- на этапе графематического анализа текст разбивается на графемы – все элементы текста (слова, знаки препинания, специальные символы и т.д.; см. п. 3.1.1.3.1);
- на этапе лексического анализа последовательность графем преобразуется в последовательность лексем, соответствующих отдельным элементам текста (см. п. 2.2.1.3.2);
- на этапе синтаксического анализа формируется синтаксическое дерево текста, определяются синтаксические структуры, их типы и взаимосвязи (см. п. 2.2.1.3.4);
- на этапе анализа кавычных выражений (названий и цитат) создаются специальные лексемы СЛОВО_В_КАВЫЧКАХ. Для прямой речи создаются граничные лексемы ЗНАК_ПРЕПИНАНИЯ; таким образом, названия отделяются от содержательного текста (см. п. 3.1.1.3.2);

- на этапе построения начальной семантической сети текста для каждой лексемы текста создается соответствующий элемент семантической сети (см. п. 3.1.1.4.2);
- на этапе фрагментации начальная семантическая сеть обрабатывается правилами нулевого слоя, в результате чего текст разбивается на фрагменты, подлежащие в дальнейшем независимой обработке в целях снижения вычислительной сложности. Процедура фрагментации эвристическая и учитывает границы предложений и параграфов, секций и т.д. (см. п. 3.1.1.4.3);
- на этапе построения начальной семантической сети фрагмента выполняется процедура, аналогичная построению начальной семантической сети для всего текста в целом (см. п. 3.1.1.4.2);
- на этапе выполнения правил первого слоя (обходом дерева правил в ширину) элементы семантической сети преобразуются в соответствии с этими правилами, в результате чего создаются семантические объекты, соответствующие соседним лексемам. Фильтрация полученной семантической сети по приоритетам и типам элементов выполняется по завершении выполнения всего дерева правил первого слоя (обязательно) или по завершении выполнения правил отдельного узла (опционально; см. п. 3.1.1.4.4);
- на этапе выполнения правил второго слоя (последовательным проходом линейной структуры правил) на основе сформированных ранее семантических объектов и добавленных синтезированных объектов создаются элементы семантической сети класса «семантическая связь». В результате фильтрации полученного множества связей формируется семантическая сеть, состоящая из семантических объектов и связей (см. п. 3.1.1.4.4);
- на этапе выполнения правил третьего слоя (последовательным проходом линейной структуры правил) создаются маркеры и маркерные связи. По завершении этапа выполняется фильтрация элементов семантической сети для отсева маркеров и маркерных связей с минимальными приоритетами (см. п. 3.1.1.4.4);
- на этапе выполнения правил четвертого слоя (обходом дерева правил в ширину) создаются сложные семантические объекты, основанные на нескольких сформированных ранее синтаксических объектах, и выполняется привязка построенных семантических объектов к синтаксическому дереву – каждому объекту сопоставляется минимальная покрывающая его синтаксическая структура. По завершении выполнения дерева правил (обязательно) и по завершении выполнения отдельных узлов (опционально) выполняется фильтрация полученной семантической сети по типам и приоритетам элементов (см. п. 3.1.1.4.4);
- на этапе объединения семантических сетей фрагментов формируется итоговая семантическая сеть из семантических сетей, полученных в результате обработки отдельных фрагментов текста (см. п. 3.1.1.4.5);
- на этапе получения результирующего множества выполняется фильтрация итоговой семантической сети по типам и приоритетам семантических объектов, а маркеры и маркерные связи преобразуются в тона семантических объектов. Таким образом, создается снимок итоговой сети в виде результирующего множества семантических объектов и их связей, допустимых с точки зрения предметной области;

- на этапе вычисления значений хэш-функций для семантических объектов рассчитываются идентифицирующие их значения предусмотренных конфигурацией предметной области хэш-функций, основанные на значениях ключевых наборов атрибутов.

Полученная в результате выполнения перечисленных операций семантическая сеть (результатирующее множество) является результатом обработки входных данных.

3.6.2 Генерация правил по шаблону

Генерация правил преобразования семантической сети в соответствии с шаблонами представляет собой вспомогательный комплекс задач ЛП «АРИОН-Лингво». Генерация правил представляет собой асинхронную процедуру и выполняется модулем `gentemplate`.

При обращении к модулю `gentemplate` передаётся описание поискового запроса прикладной информационной системы. По запросу генерируются правила предметной области, способные произвести поиск указанных объектов в разбираемых текстах.

Для обеспечения возможности генерации правил предметной области в ней должен быть создан локальный файл правил четвёртого слоя по умолчанию `rule_gentemplate.xml` следующего вида:

```
<?xml version="1.0" encoding="UTF-8"?>
<base>
  </rules4>
</base>
```

Сгенерированные правила помещаются в блоки, каждый из которых соответствует одному запросу генерирования правил. Очистка сгенерированных правил может быть совершена вызовом метода `replicateknowledge`.

Поддерживается генерация следующих запросов:

- запросы объектов;
- создание объектов;
- создание логических связей;
- ограничения на значения текстовых атрибутов `EQUAL`, `NOT_EQUAL`, `LIKE`, `NOT_LIKE`, `EMPTY`, `NOT_EMPTY`, `STARTS_WITH`, `ENDS_WITH`;
- ограничения на значения словарных атрибутов с условиями `EQUAL`, `NOT_EQUAL`;
- запросы логических связей;
- запросы ассоциативных связей;
- группы связей;
- параметры на атрибуты объектов при создании новых объектов вида `{2:АТРИБУТ}`.

3.6.3 Линеаризация (упрощение естественных языковых текстов)

Линеаризация естественных языковых текстов представляет собой вспомогательный комплекс задач ЛП «АРИОН-Лингво». Линеаризатор представляет собой специализированный механизм лингвистического процессора, обеспечивающий

преобразование текста, содержащего неявные перечисления или скрытую вариативность, в один или несколько текстов, представляющих собой явное перечисление сущностей. Для каждой линеаризованной сущности создаётся объект, связанный с исходным текстом при помощи набора экстенгов.

Основными функциями линеаризатора являются:

- расширение слов (выражений) вариантами написания с учетом вариации кодировок (пример - буквы С, О, К, Т, Е, которые одинаково выглядят в русской и английской кодировках);
- расширение слов (выражений) вариантами написания с учетом транслитерации с английского на русский язык;
- расширение слов (выражений) вариантами написания с учетом типовых опечаток/ошибок;
- расширение слов (выражений) вариантами написания с учетом фонетики (звучания);
- разбиение предложений (последовательности слов) на фрагменты с учетом синтаксического анализа.

Результатом работы линеаризатора является набор объектов заданного типа, в определённый атрибут которого (название атрибута совпадает с типом объекта) записывается линеаризованное текстовое представление. Во второй атрибут объекта записывается специализация (союз или оборот линеаризованной сущности). В опциональный атрибут с именем `ОРИГИНАЛЬНЫЙ_ТЕКСТ` записывается ненормализованный текст линеаризованной сущности.

Для каждой линеаризованной сущности генерируется множество вариантов написания следующих типов:

- замена похожих символов в различных кодировках;
- транслитерация;
- типичные опечатки из словаря;
- фонетические ошибки;
- генерация морфологических вариантов.

Варианты записываются в тэги объекта ролью `USER` и некоторым типом.

Необходимыми словарями для работы линеаризатора являются:

- англо-русский словарь транслитерации (`translit_er`);
- англо-русский словарь замены похожих символов (`translit_charset_er`);
- словарь типичных опечаток (`dict_replacement`);
- словарь негативных союзов (`dict_negativeconj`).

Подключение линеаризатора осуществляется параметром `<semanticslin>true</semanticslin>` секции `<parts>` конфигурационного файла `config_part.xml` соответствующей предметной области. К настраиваемым параметрам конфигурации линеаризатора относятся используемые им словари предметной области (см. п. 2.2.4.2.3).

3.7 Решения по составу информации, объему, способам ее организации

К информации, хранимой и используемой лингвистическим процессором в процессе его функционирования, относятся:

- конфигурационные файлы лингвистического процессора;
- конфигурационные файлы предметных областей.

3.7.1 Конфигурационные файлы лингвистического процессора

К конфигурационным файлам лингвистического процессора относятся:

- конфигурационный файл агента-планировщика;
- конфигурационный файл сервера обработки неструктурированной информации.

3.7.1.1 Конфигурационный файл агента-планировщика

Конфигурационный файл агента-планировщика `agent-config.xml` хранится в рабочем каталоге программного обеспечения ЛП и должен быть недоступен пользователям и обслуживающему персоналу ЛП за исключением технических администраторов.

К основным параметрам, определяющим характеристики функционирования агента-планировщика, относятся следующие:

- `Port` – порт для доступа к агенту-планировщику по протоколу HTTP. Указанный порт не должен использоваться другими программными продуктами и не должен быть заблокирован системным программным обеспечением;
- `DeclHost` – адрес узла, на котором должны запускаться серверы обработки неструктурированной информации. В настоящее время ЛП сконфигурирован для работы при значении этого параметра `localhost`;
- `DeclPath` – каталог, в котором расположен программный код сервера обработки неструктурированной информации. В случае, если программный код сервера расположен в рабочем каталоге агента-планировщика, значение параметра может быть пустым;
- `BasePort` – номер порта, начиная с которого будут выделяться порты запускаемым серверам обработки неструктурированной информации;
- `TraceLevel` – уровень детализации протокола обработки входных данных (уровень детализации определяет набор операций, результаты выполнения которых записываются в протокол);
- `LogFileSize` – максимальный размер протокола агента-планировщика в мегабайтах (по достижении этого размера файл протокола автоматически архивируется, а запись протокола осуществляется в новый файл);
- `LogFileRotateCount` – количество автоматически архивируемых файлов протокола (при превышении этого значения наиболее старые архивы протокола автоматически удаляются);
- `RetrainCount` – количество повторных обращений к серверу обработки неструктурированной информации при передаче запросов в случае сбоя связи;

- RetrainPause – интервал между попытками повторного обращения к серверу обработки неструктурированной информации в секундах;
- MaxTotalInstances – максимальное количество запущенных одновременно серверов обработки неструктурированной информации для одной предметной области.

Другие параметры, содержащиеся в файле agent-config.xml, являются системными и изменению не подлежат.

Пример файла agent-config.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<base>
  <config>
    <LogFileRotateCount>5</LogFileRotateCount>
    <LogFileSize>100</LogFileSize>
    <TraceLevel>ENGINE;WARNING;ERROR;INFO;PERFORM;
      DEBUG;DEBUGEXT</TraceLevel>
    <Port>8090</Port>
    <DeclPath></DeclPath>
    <DeclHost>localhost</DeclHost>
    <BasePort>10100</BasePort>
    <RetrainCount>30</RetrainCount>
    <RetrainPause>7</RetrainPause>
    <MaxTotalInstances>3</MaxTotalInstances>
    <Confinement>>false</Confinement>
    <DefaultInitKnowledge>empty</DefaultInitKnowledge>
  </config>
</base>
```

После редактирования файла agent-config.xml необходимо выполнить перезапуск агента-планировщика для того, чтобы выполненные изменения вступили в силу.

3.7.1.2 Конфигурационный файл сервера

Конфигурационный файл сервера config.xml хранится в рабочем каталоге программного обеспечения ЛП и должен быть недоступен пользователям и обслуживающему персоналу ЛП за исключением технических администраторов.

К основным параметрам, определяющим характеристики функционирования сервера, относятся следующие:

- DataPort – порт для доступа к агенту-планировщику по внутреннему протоколу на основе TCP/IP;
- Host – адрес узла агента-планировщика. В настоящее время ЛП сконфигурирован для работы при значении этого параметра localhost;

- `ValidateXmls` – параметр, определяющий необходимость контроля синтаксиса правил при помощи соответствующих схем XML;
- `BasePort` – номер порта, начиная с которого будут выделяться порты запускаемым серверам обработки неструктурированной информации;
- `TraceLevel` – уровень детализации протокола обработки входных данных (уровень детализации определяет набор операций, результаты выполнения которых записываются в протокол);
- `LogFileSize` – максимальный размер протокола агента-планировщика в мегабайтах (по достижении этого размера файл протокола автоматически архивируется, а запись протокола осуществляется в новый файл);
- `LogFileRotateCount` – количество автоматически архивируемых файлов протокола (при превышении этого значения наиболее старые архивы протокола автоматически удаляются);
- `Lang` – основной язык морфологического анализа, используемый сервером обработки неструктурированной информации;
- `DataRoot` – идентификатор предметной области, для которой сконфигурирован сервер обработки неструктурированной информации.

Другие параметры, содержащиеся в файле `config.xml`, являются системными и изменению не подлежат.

Пример файла `config.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<base>
  <config>
    <Arionmapping>>false</arionmapping>
    <ArionMapping>>false</ArionMapping>
    <ArionMappingFile>decl_mapping.xml
      </ArionMappingFile>
    <Async>>true</Async>
    <BlocksTrace>>true</BlocksTrace>
    <SecondLayerSecondPass>>true</SecondLayerSecondPass>
    <ChangesTrace>>false</ChangesTrace>
    <CleanBeforeRules>>true</CleanBeforeRules>
    <CommaTolerance>3</CommaTolerance>
    <CycleThreshold>40</CycleThreshold>
    <DamagedText>>true</DamagedText>
    <DataPort>10100</DataPort>
    <DataRoot>ARION-FS</DataRoot>
    <Host>localhost</Host>
    <DebugOut>>true</DebugOut>
    <DeleteBaseLexems>>true</DeleteBaseLexems>
    <EnableLoops>>true</EnableLoops>
    <FailIfFamilyIsPatr>>true</FailIfFamilyIsPatr>
```

```
<Filter>filter.xml</Filter>
<FilterEmptyVerbsAsNear>>false</FilterEmptyVerbsAsNear>
<FilterLinks>>true</FilterLinks>
<FioRollback>>true</FioRollback>
<InnerSentenceTolerance>0</InnerSentenceTolerance>
<KeepLostLinks>>false</KeepLostLinks>
<Lang>russian</Lang>
<LearningEstimateTime>>false</LearningEstimateTime>
<LogFileRotateCount>5</LogFileRotateCount>
<LogFileSize>100</LogFileSize>
<LogPort>10001</LogPort>
<LogSentenceEdges>>false</LogSentenceEdges>
<ModulesPath>modules</ModulesPath>
<MorphOut>>true</MorphOut>
<NoComponentPrio>>true</NoComponentPrio>
<NoErrorMsg>>false</NoErrorMsg>
<NoResultTypesFilter>>false</NoResultTypesFilter>
<OldStyleLinks>>false</OldStyleLinks>
<OldStyleSecondGrid>>false</OldStyleSecondGrid>
<Output>output.xml</Output>
<PreprocessorFile></PreprocessorFile>
<QuotedWordThreshold>5</QuotedWordThreshold>
<QuoteThreshold>14</QuoteThreshold>
<RecorderFile></RecorderFile>
<RemoveFilterDups>>true</RemoveFilterDups>
<RmlPath>Rml</RmlPath>
<Rules>rules.xml</Rules>
<Rules2>rules2.xml</Rules2>
<RulesRoot>rules</RulesRoot>
<SentFiltering>>false</SentFiltering>
<SplitText>300</SplitText>
<SplitThreshold>0</SplitThreshold>
<TimeEstimateFile>timing.xml</TimeEstimateFile>
<TraceLevel>ENGINE;WARNING;ERROR;INFO;PERFORM;DEBUG;
    DEBUGEXT</TraceLevel>
<TwoStagesFilter>>true</TwoStagesFilter>
<UserRoot>user</UserRoot>
<ValidateXmls>>true</ValidateXmls>

</config>
</base>
```

После редактирования файла `config.xml` необходимо произвести перезапуск выполняющихся серверов обработки неструктурированной информации для их инициализации в соответствии с выполненными изменениями.

3.7.2 Конфигурационные файлы предметных областей

К конфигурационным файлам предметных областей относятся:

- описание состава конфигурационных файлов;
- общие настройки предметной области;
- словари предметной области;
- правила предметной области;
- манифесты внешних библиотек.

Конфигурационные файлы предметных областей могут публиковаться и храниться двумя способами:

- в отдельном каталоге файловой системы;
- в архиве формата ZIP с контрольной суммой.

В случае публикации конфигурации предметной области в виде архива в состав настроек включаются архив конфигурационных файлов `rules-<дескриптор>.zip` и файл контрольной суммы `rules-<дескриптор>.sha`, где `<дескриптор>` – идентификатор предметной области. Контрольная сумма вычисляется в соответствии с алгоритмом SHA-512. Верификация контрольной суммы конфигурации предметной области выполняется в процессе инициализации сервера ЛП.

3.7.2.1 Описание состава конфигурационных файлов

Основным конфигурационным файлом предметной области, содержащим описание остальных конфигурационных файлов, является файл `config-part.xml`, размещенный в каталоге предметной области (архиве). В этом файле описываются основные параметры предметной области (такие, как язык и альтернативный идентификатор предметной области), а также перечисляются все конфигурационные файлы, относящиеся к предметной области.

Язык предметной области задается в файле `config-part.xml` тэгом `<language>`, допустимыми значениями которого являются строки `"russian"` или `"english"`. Значение параметра по умолчанию – `"russian"`.

Альтернативный (например, кириллический) идентификатор предметной области может быть задан при помощи тэга `<name>`. Если такой идентификатор явно не указан, то в качестве идентификатора используется название каталога с конфигурацией предметной области.

Предельное время обработки одного запроса указывается при помощи тэга `<timelimit>`. При этом заданное в файле `config-part.xml` предельное время обработки измеряется в минутах.

Если правила предметной области разработаны таким образом, что могут быть использованы для обработки DOM-документов, в файле `config-part.xml` тэг `<dommed>` должен иметь значение `"true"`. Файлы, содержащие настройки рекодера и препроцессора, могут задаваться необязательными тэгами `<recoder>` и `<preprocessor>`.

Конфигурационные файлы предметной области указываются в файле `config-part.xml` в соответствующих секциях. Формат секции имеет вид:

```
<ТИП_ДАНЫХ name="ПСЕВДОНИМ">
  <item on="true|false">ПУТЬ_И_НАЗВАНИЕ_ФАЙЛА</item>
  ...
  <item on="true|false">ПУТЬ_И_НАЗВАНИЕ_ФАЙЛА</item>
</ТИП_ДАНЫХ>
```

Здесь ТИП_ДАНЫХ указывает назначение разбора конфигурационных файлов, перечисленных в секции; ПСЕВДОНИМ – общее название именного (см. ниже) набора конфигурационных файлов для использования в файлах манифестов внешних библиотек. Пути к файлам ПУТЬ_И_НАЗВАНИЕ_ФАЙЛА, определенные секцией, рассматриваются как локальные внутри каталога предметной области. Отдельные конфигурационные файлы, названия которых перечислены в секции, можно подключать, установив значение параметра `on` в `"true"`.

Назначение набора конфигурационных файлов определяется названием тэга ТИП_ДАНЫХ. Допустимые названия тэга следующие:

- `rules0`;
- `rules1`;
- `rules2`;
- `rules3`;
- `rules4`;
- `delims` – (словарь разделителей лексем);
- `dict` (словари категорий);
- `tones` (словари тональностей);
- `translit` (словари транслитерации).

Наборы конфигурационных файлов с назначением `dict`, `tones`, `translit` являются именованными, т.е. таких наборов может быть несколько в пределах одной предметной области. В этом случае наборы различаются по значению параметра `name`. Остальные наборы конфигурационных файлов относятся к неименным, т.е. все упоминаемые в таких секциях файлы объединяются в списки и используются только вместе. Параметр `name` таких секций игнорируется.

Параметр `name` именованных наборов конфигурационных файлов позволяет функциям внешних библиотек обращаться к конкретному словарю по его идентификатору. Для этого значение параметра `name` указывается в манифесте соответствующей внешней библиотеки.

Пример файла `config_part.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<parts>
  <language>russian</language>
  <name>MyKnowledge</name>
```

```
<timelimit>5</timelimit>
<dommed>>false</dommed>

<recoder>recoder.xml</recoder>
<preprocessor>preprocessor.xml</preprocessor>

<rules0>
  <item on="false">rules0.xml</item>
</rules0>

<rules>
  <item on="true">rules.xml</item>
  <item on="false">../common/object_NP_MAKE.xml</item>
</rules>

<rules2>
  <item on="true">rules2.xml</item>
</rules2>

<rules4>
  <item on="true">rules4.xml</item>
</rules4>

<dict name="dict_COMMON_ABBR.xml">
  <item on="true">dict_COMMON_ABBR.xml</item>
</dict>

<dict name="dict_COMMON_ABBR2.xml">
  <item on="true">dict_COMMON_ABBR2.xml</item>
</dict>

<dict name="dict_NP_I_MAKE.xml">
  <item on="true">dict_NP_I_MAKE.xml</item>
</dict>

<dict name="dict_NP_I.xml">
  <item on="true">dict_NP_I.xml</item>
</dict>

<dict name="special">
  <item on="true">dict_special.xml</item>
</dict>

<tones name="mytone">
  <item on="false">foo.xml</item>
</tones>

<translit name="intuit">
  <item on="true">translit_intuit.xml</item>
</translit>

<translit name="reverse">
  <item on="true">translit_reverse.xml</item>
</translit>

</parts>
```

После редактирования файла `config_part.xml` необходимо выполнить перезапуск выполняющихся серверов обработки неструктурированной информации,

настроенных на текущую предметную область, для того, чтобы выполненные изменения вступили в силу.

3.7.2.2 Общие настройки предметной области

К общим настройкам предметной области относятся:

- перечень типов объектов и связей предметной области;
- настройки хэширования семантических объектов.

Перечень типов объектов и связей предметной области является основой для процедур фильтрации семантической сети на ряде этапов обработки входных данных. В перечне указываются типы семантических объектов, списки их атрибутов, ключевые наборы атрибутов, а также типы семантических связей. Все объекты, атрибуты и связи, не соответствующие приведенному перечню, исключаются из результирующего множества.

Настройки хэширования семантических объектов обеспечивают возможность вычисления значений хэш-функций для последующей идентификации объектов средствами прикладной информационной системы. К настройкам хэширования относятся состав и идентификаторы ключевых наборов атрибутов.

Общие настройки предметной области (перечень типов объектов и связей, а также настройки хэширования объектов) содержатся в файле `result_types.xml`, имеющем формат XML. При описании общих настроек предметной области используются следующие конструкции языка XML:

- секция `<knowledgemap>`, содержащая перечень типов семантических объектов (тэг `<objects>`) и перечень типов семантических связей (тэг `<links>`);
- секция `<objects>`, содержащая список типов семантических объектов (тэги `<object>`);
- секция `<object>`, содержащая описание одного типа семантических объектов, задаваемого его названием (текстовое значение тэга `<type>`) списком атрибутов (тэг `<attrs>`) и списком ключевых наборов атрибутов (тэг `<keysets>`);
- секция `<attrs>`, содержащая список атрибутов для объектов данного типа (тэги `<attr>`);
- тэг `<attr>`, фиксирующий необходимый атрибут семантического объекта, принадлежащего заданному типу. Название атрибута определяется текстовым значением тэга `<attr>`;
- секция `<keysets>`, содержащая список ключевых наборов атрибутов для объектов данного типа (тэги `<keyset>`);
- тэг `<keyset>`, определяющий ключевой набор атрибутов семантического объекта, принадлежащего заданному типу. Ключевой набор атрибутов задаётся идентификатором (тэг `<name>`) и списком атрибутов (тэги `<attr>` в секции `<attrs>`);
- секция `<links>`, содержащая список типов семантических связей (тэги `<link>`);
- тэг `<link>`, описывающий один тип семантической связи по её названию (текстовое значение тэга `<type>`).

Примером содержимого файла общих настроек предметной области result_types.xml является файл следующего вида:

```
<?xml version="1.0" encoding="UTF-8" ?>
<base>
  <knowledgemap>
    <objects>
      <object>
        <type>ФИЗ_ЛИЦО</type>
        <attrs>
          <attr>ВОЗРАСТ</attr>
          <attr>ДАТА_РОЖДЕНИЯ</attr>
          <attr>ИМЯ</attr>
          <attr>ИНН</attr>
          <attr>КЛИЧКА</attr>
          <attr>НОМЕР_ПАСПОРТА</attr>
          <attr>НОМЕР_УДОСТОВЕРЕНИЯ</attr>
          <attr>ОТЧЕСТВО</attr>
          <attr>КЛАССИФИКАТОР</attr>
          <attr>ФАМИЛИЯ</attr>
          <attr>ПОЛ</attr>
        </attrs>
        <keysets>
          <keyset nameHash="true">
            <name>Набор1</name>
            <attrs>
              <attr>ИНН</attr>
              <attr>НОМЕР_ПАСПОРТА</attr>
            </attrs>
          </keyset>
          <keyset nameHash="false"
            sourceHash="true">
            <name>Набор2</name>
            <attrs>
              <attr>ФАМИЛИЯ</attr>
              <attr>ИМЯ</attr>
              <attr>ОТЧЕСТВО</attr>
            </attrs>
          </keyset>
        </keysets>
      </object>
      <object>
        <type>ОРГАНИЗАЦИЯ</type>
```

```

        <attrs>
            <attr>НАИМЕНОВАНИЕ_ПОЛНОЕ</attr>
            <attr>НАИМЕНОВАНИЕ_КРАТКОЕ</attr>
            <attr>ВИД_ДЕЯТЕЛЬНОСТИ</attr>
            <attr>ТИП</attr>
            <attr>ВЕС</attr>
        </attrs>
    </object>
    <object>
        <type>ДАТА</type>
        <attrs>
            <attr>ПРИМЕЧАНИЕ</attr>
            <attr>ДЕНЬ</attr>
            <attr>МЕСЯЦ</attr>
            <attr>ГОД</attr>
            <attr>ISO</attr>
        </attrs>
    </object>
</objects>
<links>
    <link>
        <type>СВЯЗЬ_ФЛ_ДАТА</type>
    </link>
    <link>
        <type>СВЯЗЬ_ФЛ_ОРГАНИЗАЦИЯ</type>
    </link>
    <link>
        <type>СВЯЗЬ_ОРГАНИЗАЦИЯ_ДАТА</type>
    </link>
</links>
</knowledgemap>
</base>

```

Файл `result_types.xml` должен находиться в каталоге с описанием конфигурации предметной области.

3.7.2.3 Словари предметной области

Словарь предметной области ЛП «АРИОН-Лингво» представляет собой совокупность сведений о соответствии между семантическими доменами, т.е. о сопоставлении одних терминов другим по принципу «один ко многим» или «многие к одному». С технической точки зрения словари представляют собой файлы формата XML, в которых соотношение между семантическими доменами задается при помощи секций следующего вида (приведены условные названия тэгов; формат конкретных словарей отличается от приведенного обобщенного описания):

```

<item>
    <src>НАЗВАНИЕ_ДОМЕНА_1</src>
    ...

```

```
<src>НАЗВАНИЕ_ДОМЕНА_N</src>
<dest>НАЗВАНИЕ_ДОМЕНА_N+1</dest>
...
<dest>НАЗВАНИЕ_ДОМЕНА_N+M</dest>
</item>
```

Здесь тэги `<src>` определяют семантические домены нижнего уровня, которые связываются с семантическими доменами верхнего уровня, задаваемыми тэгами `<dest>`. В каждой секции `<item>` должны присутствовать как минимум один тэг `<src>` и как минимум один тэг `<dest>`, при этом в рамках секции только для одного вида тэгов допустимо множественное вхождение.

Совокупность секций `<item>`, аналогичных с точки зрения соотношения количества тэгов `<src>` и `<dest>` в них, образует словарь предметной области. По типам входящих в них секций словари подразделяются на два класса:

- словари типа «один ко многим», секции `<item>` которых содержат один тэг `<src>` и один или более тэгов `<dest>`. Основное назначение словарей типа «один ко многим» – определение всех возможных семантических доменов (`<dest>`) по заданному ключу (`<src>`);
- словари типа «многие к одному», секции `<item>` которых содержат один или более тэгов `<src>` и один тэг `<dest>`. Основное назначение таких словарей – определение семантического домена (`<dest>`), фиксирующего тип заданного ключа (`<src>`).

Примером словаря типа «один ко многим» может служить словарь расшифровок, содержащий секции следующего вида:

```
<item>
  <src>Г</src>
  <dest>ГОД</dest>
  <dest>ГОРОД</dest>
  <dest>ГОСПОДИН</dest>
</item>
```

Примером словаря типа «многие к одному» может служить словарь категорий, содержащий секции следующего вида:

```
<item>
  <src>АВЕНЮ</src>
  <src>АЛЛЕЯ</src>
  <src>БУЛЬВАР</src>
  <src>ВАЛ</src>
  <src>ГОРА</src>
  <src>КОЛЬЦО</src>
  <src>НАБЕРЕЖНАЯ</src>
  <src>ПЕРЕУЛОК</src>
  <src>ПЛОЩАДЬ</src>
  <src>ПРОЕЗД</src>
  <src>ПРОСЕКА</src>
  <src>ПРОСПЕКТ</src>
  <src>ТУПИК</src>
```

```
<src>УЛИЦА</src>
<src>ХОЛМЫ</src>
<src>ШОССЕ</src>
<dest>ТИП_УЛИЦЫ</dest>
</item>
```

Словарь может быть интерпретирован различными способами (см. ниже), за исключением словаря тональностей, который обладает особым форматом представления и наряду с указанием соответствия семантических доменов задает вес этого соответствия.

Примером словаря предметной области является словарь транслитераций – словарь, содержащий правила транслитерации русскоязычного текста. Такой словарь содержит отображение символа на множество вариантов транслитерации этого символа:

```
<?xml version="1.0" encoding="utf-8"?>
<base>
  <translit friendlyName="foo">
    <entry>
      <from>a</from>
      <to>a</to>
      <to>я</to>
    </entry>
    <entry>
      <from>b</from>
      <to>б</to>
    </entry>
    <entry>
      <from>c</from>
      <to>ц</to>
      <to>ч</to>
    </entry>
  </translit>
</base>
```

Другим примером словаря предметной области является словарь разделителей лексем, который позволяет задать подстроки, по которым будут разбиваться лексемы на этапе графематического анализа. Словарь содержит перечисление возможных разделителей:

```
<?xml version="1.0" encoding="UTF-8"?>
<base>
  <delimiters friendlyname="РАЗДЕЛИТЕЛИ_ВАЛЮТ">
    <item>USD</item>
    <item>RUR</item>
    <item>EUR</item>
  </delimiters>
```

```
</base>
```

Особый класс словарей образуют словари тональностей, которые представляют собой расширение обычных словарей, позволяющее указывать веса отношений между семантическими доменами. Такие словари используются для определения по термину типа его тональности и веса этого термина в рамках тональности.

Пример словаря тональностей:

```
?xml version="1.0" encoding="UTF-8"?>
<base>
  <tones friendlyName="Специальный словарь">
    <tone>
      <type>ОЦЕНКА</type>
      <values>
        <value weight="100">ПРЕВОСХОДНО</value>
        <value weight="50">ЧУДЕСНО</value>
        <value weight="20">ХОРОШО</value>
        <value weight="-40">НЕЙТРАЛЬНО</value>
        <value weight="-200">ПЛОХО</value>
        <value weight="-999">УЖАСНО</value>
      </values>
    </tone>
    <tone>
      <type>ОТНОШЕНИЕ_К_ТОРГОВЛЕ</type>
      <values>
        <value weight="100">ГАСТРОНОМ</value>
        <value weight="100">ГИПЕРМАРКЕТ</value>
        <value weight="100">ТОРГОВЫЙ ДОМ</value>
        <value weight="100">УНИВЕРМАГ</value>
        <value weight="90">ЛАРЁК</value>
        <value weight="-100">АКАДЕМИЯ НАУК</value>
      </values>
    </tone>
  </tones>
</base>
```

Словари предметной области перечисляются в файле конфигурации предметной области `config_part.xml` при помощи секций следующего вида:

```
<ТИП_ДАННЫХ name="ПСЕВДОНИМ">
  <item on="true">ПУТЬ_И_НАЗВАНИЕ_ФАЙЛА</item>
</ТИП_ДАННЫХ>
```

Здесь `ТИП_ДАННЫХ` – тип секции, определяющий тип подключаемого словаря, `ПСЕВДОНИМ` – название словаря, используемое внутренними компонентами ЛП для

обращения к словарю, ПУТЬ_И_НАЗВАНИЕ_ФАЙЛА – название XML-файла, содержащего словарь.

3.7.2.4 Правила предметной области

Правила предметной области (правила преобразования семантической сети) являются основным средством настройки ЛП «АРИОН-Лингво» в соответствии с особенностями конкретной предметной области. Правила содержат условия отбора (секция условий) и инструкции по преобразованию (секция действий), применяемые к элементам семантической сети.

Общий синтаксис правила преобразования семантической сети:

```
<RULE priority="ПРИОРИТЕТ" nospace="ФЛАГ_РАЗРЫВА"  
      filterwithcontexts="ФЛАГ_ФИЛЬТРА">  
  
  УСЛОВИЕ_0  
  УСЛОВИЕ_1  
  ...  
  УСЛОВИЕ_M  
  ДЕЙСТВИЕ_M+1  
  ДЕЙСТВИЕ_M+2  
  ...  
  ДЕЙСТВИЕ_M+K  
  
</RULE>
```

Выполнение правила включает два этапа:

- распознавание. На этом этапе происходит последовательная проверка условий. Если в результате проверки условия выбираются несколько элементов семантической сети, удовлетворяющих условию, то происходит соответствующее ветвление процесса выполнения правила (каждая ветвь соответствует некоторой цепочке вывода). Если условие не выполняется, текущая ветвь выполнения правила прерывается;
- реагирование. На этом этапе происходит последовательное выполнение операций по модификации семантической сети, предусмотренных секцией действий. Ветвления процесса выполнения правила на этапе реагирования не происходит.

Иными словами, в процессе выполнения правила в первую очередь проверяются условия отбора объектов. Для каждого обнаруженного в результате проверки набора объектов создается ветвь выполнения правила, в рамках которой значения переменных определяются этим набором объектов. Если очередному условию правила не удовлетворяет ни один объект семантической сети, то выполнение соответствующей ветви прерывается. Если для некоторой ветви выполнения правила (набора объектов) справедливы все условия этого правила, то к этому набору объектов применяются инструкции (операторы) секции действий. Таким образом, каждое правило применяется для всех наборов объектов, удовлетворяющих его секции условий.

Правила преобразования семантической сети подразделяются на слои, каждый из которых предназначен для выполнения преобразований определенного вида. Ограничения по применению правил на различных слоях не предусмотрены. В целях снижения

ресурсоёмкости алгоритмов обработки данных при выполнении правил первого слоя набор элементов семантической сети считается удовлетворяющим секции условий только в том случае, если этот набор соответствует непрерывному фрагменту лексической структуры текста. Указанная особенность реализована средствами интерпретатора и не устанавливает ограничений на синтаксис и семантику правил преобразования сети.

В секциях условий и секциях действий правил возможно обращение к словарям предметной области и функциям внешних библиотек. Правила формулируются на специальном непроцедурном языке.

3.7.2.5 Манифесты внешних библиотек

Внешние библиотеки предназначены для хранения и выполнения прикладных функций, используемых правилами преобразования семантической сети для проверки условий и модификации данных. Манифест внешней библиотеки содержит перечень словарей, которые используются функциями этой библиотеки, что позволяет серверу обработки неструктурированной информации предоставить функциям необходимые для выполнения данные. Манифест внешней библиотеки не является обязательным; это означает, что функции такой библиотеки не нуждаются в использовании словарей.

Файл манифеста внешней библиотеки имеет название, совпадающее с названием библиотеки, но отличающееся расширением `xml`. Формат файла манифеста имеет вид:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest>
  <vocab>
    <name>abbr</name>
    <path>dict_COMMON_ABBR.xml</path>
  </vocab>
  <category>
    <name>streets</name>
    <path>dict_STREETS.xml</path>
  </category>
</manifest>
```

Здесь `vocab` и `category` – типы интерпретации словарей (см. п. 3.1.1.5.2); допустимыми типами интерпретации являются также `tone` и `translit`. Тэг `<name>` содержит внутреннее для библиотеки название словаря, по которому функции этой библиотеки обращаются к этому словарю. Тэг `<path>` содержит идентификатор словаря, который должен точно совпадать с названием именного набора конфигурационных файлов, соответствующего этому словарю, в файле `config_part.xml`.

Любой словарь предметной области может быть интерпретирован внешней библиотекой как словарь «один ко многим» или как словарь «многие к одному» в зависимости от того, в каком образом он описан в манифесте этой библиотеки.

3.8 Решения по составу программных средств

Состав программных средств ЛП определяется используемой платформой (Win32, Linux, MAC OS X) и конфигурацией подсистем ЛП.

Запуск программного обеспечения ЛП осуществляется в зависимости от используемой платформы:

- в среде Win32 ЛП может быть сконфигурирован как служба или как приложение. В первом случае инсталлятор автоматически регистрирует и запускает ЛП (опционально). Принудительный запуск и остановка службы ЛП возможны через панель администрирования или посредством скриптов `start.bat` и `stop.bat` соответственно. Во втором случае запуск ЛП осуществляется как запуск обычного приложения, но в командной строке указывается параметр `/simple`;
- в UNIX-подобных операционных системах запуск ЛП производится путём запуска демона `bin/arionlingvo-agent`. Остановка ЛП выполняется командой `bin/arionlingvo-agent -x`. Для запуска демона можно также воспользоваться скриптом `bin/start.sh`.

Дополнительные параметры запуска сервера ЛП (указываются в командной строке):

- `-d, --dir=PATH` – рабочий каталог (по умолчанию – каталог запуска);
- `-p, --portdata=PORT` – явное указание значения `dataport` (см. описание файла `config.xml`);
- `-l, --portlog=PORT` – явное указание значения `logport` (см. описание файла `config.xml`);
- `-k, --knowledge=KNOWLEDGE` – явное указание значения `knowledgebase` (см. описание файла `config.xml`);
- `-s, --suffix=FILENAME` – суффикс, добавляемый к названиям файлов протоколов ЛП;
- `-c, -` – консольный режим запуска ЛП;
- `-h, --help` – вывод справочной информации о ЛП.
- `-i, --install` – установка ЛП в качестве службы (только Win32);
- `-u, --uninstall` – отмена установки ЛП в качестве службы (только Win32).

Запуск ЛП без параметров приводит к запуску агента-планировщика в виде сервиса/демона, с параметром `-c` – к запуску агента как консольного приложения.

Минимальные требования к операционной системе сервера ЛП:

- Windows XP или Windows Server 2003;
- Linux с версией ядра 2.6;
- Mac OS X версии 10.5.

3.8.1 Особенности обработки русскоязычных текстов

При обработке русскоязычных текстов ЛП использует стандартные обозначения для основных типов элементов лексической и синтаксической структуры. К таким обозначениям относятся базовые лексемы, типы синтаксических клауз и типы синтаксических групп.

Конфиденциально

Перечень базовых лексем:

Тип	Структура
СЛОВО	два аргумента: значение и флаг регистра. Определяет слово из русских букв.
WORD	два аргумента: значение и флаг регистра. Определяет слово из латинских букв.
СМЕШАННОЕ_СЛОВО	два аргумента: значение и флаг регистра. Определяет слово из русских и латинских букв.
СЛОВО_В_КАВЫЧКАХ	один аргумент: значение. Определяет слово в кавычках.
БЛОК_ЧИСЕЛ	один аргумент: значение. Определяет последовательность цифр.
БЛОК_БУКВЕННО_ЦИФРОВОЙ	один аргумент: значение. Определяет слово из букв и цифр.
ЗНАК_ПУНКТУАЦИИ	один аргумент: тип знака. Определяет знак пунктуации.
СПЕЦИАЛЬНЫЙ_ЗНАК	один аргумент: тип знака. (КОНЕЦ_ПРЕДЛОЖЕНИЯ, НОВАЯ_СТРОКА)

Для каждой лексемы устанавливается флаг регистра, принимающий следующие значения:

- Up (слово из заглавных букв);
- Lw (слово из строчных букв);
- UpLw (слово с заглавной буквы);
- UpLwUp (слово со строчной буквы, но хотя бы с одной заглавной).

Все базовые лексемы имеют длину 1, кроме СПЕЦИАЛЬНЫЙ_ЗНАК (длина 0 - что позволяет игнорировать наличие этой лексемы правилами). В процессе анализа текста создаются лексемы, соответствующие словам. Такие лексемы также имеют длину 1.

Перечень типов синтаксических клауз:

Тип	Сокращенное название	Пример
Фрагмент с личной формой глагола	ГЛ_ЛИЧН	Я иду домой
Деепричастный оборот	ДПР	идя домой
Фрагмент с кратким причастием	КР_ПРЧ	Она снята
Фрагмент с кратким прилагательным	КР_ПРИЛ	Она красива
Фрагмент с предикативом	ПРЕДК	мне интересно
Причастный оборот	ПРЧ	дом, стоявший на холме
Фрагмент с инфинитивом	ИНФ	чтобы лучше жить

Вводный оборот	ВВОД	на самом деле
Фрагмент с тире	ТИРЕ	Моя сестра - комсомолка.
Необособленное согласованное определение в препозиции	НСО	для известного всем дворника
Фрагмент со сравнительным прилагательным	СРАВН	идя домой

Перечень типов синтаксических групп:

Тип	Сокращенное название	Пример
Количественная группа	КОЛИЧ	двадцать восемь
Последовательность чисел вперемешку со знаками препинания	КОЛИЧ	12,2
Существительное из заданного перечня + числовой идентификатор	СУЩ-ЧИСЛ	статья 123
Правила для построения ФИО (используются морфологические пометы о том, что данное слово может быть именем)	ФИО	Петров Петр Владимирович
Слова степени (типа "очень") с группой прилагательного или причастия	НАР_ПРИЛ	очень красивый
Однородные прилагательные	ОДНОР_ПРИЛ	первой и единственной
Однородные наречия	ОДНОР_НАР	долго иль коротко
Однородные инфинитивы	ОДНОР_ИНФ	стоять или лежать
Однородные прилагательные сравнительной степени	ОДНОР_ПРИЛ	красивее и моложе
Группы даты	ДАТА	август 1968 года, 12 июня 99 г. и т.д.
Группа временных отрезков	СЛОЖ_ПГ	С первого августа по двадцатое сентября
Аналитическая форма сравнительной степени прил. или наречия	СРАВН-СТЕПЕНЬ	гораздо сильнее
Наречие + глагол	НАРЕЧ-ГЛАГОЛ	злостно нарушать
Одно или несколько прилагательных, согласованных по роду, числу и падежу со стоящим сразу после них существительным.	ПРИЛ-СУЩ	длинная унылая дорога
Наречное числительное + ИГ (рд мн)	НАР-ЧИСЛ-СУЩ	много очень простых ребят
Числительное + ИГ	ЧИСЛ-СУЩ	сорок восемь попугаев
Генитивная пара	ГЕНИТ_ИГ	рука Москвы
Предложная группа	ПГ	на холме
Однородные ИГ	ОДНОР_ИГ	мама и папа

Тип	Сокращенное название	Пример
Отрицание + глагольная форма	ОТР_ФОРМА	не любить
Глагольная форма+контактное прямое дополнение	ПРЯМ_ДОП	рубить дрова
Группа электронного адреса	ЭЛ_АДРЕС	www.dialing.ru
Глагольная форма+контактный инфинитив	ГЛАГ_ИНФ	пойти выпить
Подлежащее	ПОДЛ	я пошел
Сказуемое	СКАЗ	я пошел

4 МЕРОПРИЯТИЯ ПО ПОДГОТОВКЕ К ВВОДУ В ДЕЙСТВИЕ

4.1 Мероприятия по приведению информации к виду, пригодному для обработки

Приведение информации к виду, пригодному для автоматической обработки ЛПП «АРИОН-Лингво», заключается в преобразовании информационных сообщений к одному из трех форматов представления исходных данных, поддерживаемых лингвистическим процессором. Для выполнения такого преобразования необходимо выполнить следующие процедуры:

- для преобразования HTML-документа в форму DOM-документа запустить Internet Explorer и воспользоваться функциями соответствующего плагина или передать HTML-документы специальному средству их пакетной обработки (см. п. 3.2.1.3);
- для преобразования полнотекстового документа в форму частично структурированного документа выделить в составе документа информационные объекты в соответствии с перечнем типов объектов предметной области и сформировать на основе выделенных объектов и исходного текста документа соответствующие секции частично структурированного документа (см. описание формата частично структурированного документа в п. 3.2.1.2);
- для преобразования произвольного документа в форму плоского текста выполнить преобразование при помощи соответствующего текстового редактора (см. описание дополнительных параметров в п. 3.2.1.1).

Преобразование информационных сообщений к виду, пригодному для обработки средствами ЛПП «АРИОН-Лингво» может осуществляться как в пакетном режиме для последующей массовой обработки, так и индивидуально для каждого информационного сообщения в зависимости от логики функционирования автоматизированной информационной системы, в состав которой входит лингвистический процессор.

4.2 Мероприятия по обучению и проверке квалификации персонала

К квалификации персонала, обеспечивающего функционирование компонентов ЛПП «АРИОН-Лингво», предъявляются следующие основные требования:

- технический администратор должен:
 - знать: технологию функционирования ЛПП; назначение параметров конфигурации компонентов ЛПП; особенности функционирования и настройки компонентов ЛПП и системного программного обеспечения.
 - уметь: самостоятельно осуществлять установку, настройку и восстановление работоспособности ЛПП и системного программного обеспечения; самостоятельно изменять конфигурационные файлы ЛПП для повышения его эксплуатационных характеристик; диагностировать и устранять неполадки в работе программного и технического обеспечения.
- инженер по знаниям должен:
 - знать: особенности модели предметной области, используемой ЛПП; технологию функционирования компонентов ЛПП, осуществляющих

непосредственную обработку данных; форматы входных и выходных данных ЛП; структуру файлов конфигурации предметной области; назначение параметров конфигурации предметной области; особенности настройки ЛП на конкретные предметные области.

уметь: самостоятельно составлять и корректировать перечень допустимых типов объектов и связей конкретной предметной области; самостоятельно составлять и корректировать наборы правил, соответствующие конкретной предметной области; самостоятельно формировать и корректировать файлы манифестов внешних библиотек; обнаруживать и устранять ошибки в части конфигурации предметных областей.

– лингвист должен:

знать: типы словарей, используемые ЛП; технологию использования различных типов словарей компонентами ЛП; форматы представления словарей;

уметь: самостоятельно разрабатывать и корректировать словари для настройки ЛП на конкретную предметную область; самостоятельно включать разработанные словари в состав конфигурационных файлов предметной области; обнаруживать и исправлять ошибки в используемых словарях предметных областей.

Перечень мероприятий по обучению и проверке квалификации персонала определяется штатной структурой подразделения, обеспечивающего функционирование ЛП «АРИОН-Лингво» и формируется на этапе ввода системы в опытную эксплуатацию.

4.3 Мероприятия по созданию необходимых подразделений и рабочих мест

В целях обеспечения непрерывного функционирования ЛП «АРИОН-Лингво» необходимо включить в состав подразделения, обеспечивающего функционирование автоматизированной информационной системы, дополнительные штатные единицы администраторов, инженеров по знаниям и лингвистов в следующей зависимости от ежедневного количества информационных сообщений, подлежащих обработке ЛП:

Количество сообщений (в день)	100 – 1000	1000 – 10000	10000 – 100000
Администраторов	1	1	2
Инженеров по знаниям	1	2	4
Лингвистов	1	2	4

Организационно обслуживающий персонал ЛП «АРИОН-Лингво» может быть выделен в административную группу в составе подразделения, основной функциональной обязанностью которого является поддержание работоспособности системы.

ПРИЛОЖЕНИЕ 1 ОПИСАНИЕ МОДУЛЯ ПАКЕТНОЙ ОБРАБОТКИ ВХОДНЫХ ДАННЫХ DPS

Общие сведения

Модуль DPS представляет собой программное обеспечение, запускаемое из командной строки и предназначенное для управления лингвистическим процессором «АРИОН-Лингво» и пакетной обработки входных данных.

Программный код и конфигурация DPS располагается в каталоге файловой системы dps. Модуль реализован на платформе Java и требует для выполнения JRE версии 1.5 и выше.

Параметры модуля

Запуск модуля DPS осуществляется в командной строке или командном файле инструкцией следующего вида:

```
java -Dfile.encoding=utf8 -jar dps.jar <действие> <аргументы>
```

Здесь <действие> – обозначение операции, для выполнения которой используется DPS; <аргументы> – параметры операции.

К служебным операциям, которые могут быть выполнены модулем DPS, относятся:

Действия	Описание
info	– получение информации о лингвистическом процессоре
reload	– принудительная перезагрузка лингвистического процессора
shutdown	– принудительное завершение работы лингвистического процессора

К операциям по обработке отдельных файлов входных данных, которые могут быть выполнены модулем DPS, относятся:

Действия	Описание
text	– обработка отдельного текстового файла, название которого указывается в последующем обязательном параметре
domtask	– выполнение отдельного задания по обработке DOM-документа. В последующем обязательном параметре указывается путь к заданию
webarchive	– обработка web-архива в соответствии с шаблонными заданиями. Обязательные параметры – путь к каталогу с web-архивом и путь к каталогу с заданиями

`webarchivelite` – редуцированная обработка web-архива. Отличие от операции `webarchive` заключается в том, что обработка выполняется до этапа получения DOM-документов, которые являются выходными данными

В случае обработки web-архива результаты сохраняются в папке web-архива в виде файлов с расширением `.dal`, представляющих собой XML-документ в базовом формате результирующего множества (при этом в режиме `webarchive` в секции `<metadata>` создаётся тэг `<rootpath>`, в котором сохраняется информация об обработанном фрагменте web-страницы).

К операциям по пакетной обработке входных данных, которые могут быть выполнены модулем DPS, относятся:

Действия	Описание
<code>plain</code>	– обработка подборки текстовых файлов. Обязательный параметр – путь к каталогу файловой системы, в котором расположена подборка
<code>ariondoc</code>	– обработка подборки частично структурированных документов. Обязательный параметр – путь к каталогу файловой системы, в котором расположена подборка
<code>mdsdoc</code>	– обработка подборки файлов базового формата представления результирующего множества. Обязательный параметр – путь к каталогу файловой системы, в котором расположена подборка
<code>domdoc</code>	– обработка подборки DOM-документов. Обязательный параметр – путь к каталогу файловой системы, в котором расположена подборка

Если в конфигурационном файле DPS определён параметр `inputregex`, то его значение используется как маска для отбора файлов в заданном каталоге. Результаты разбора сохраняются с учетом значений параметров конфигурации `xmls` и `xmlsnear` с добавлением к имени файла расширения `.xml`.

Параметры конфигурации

Параметры конфигурации модуля DPS определяются в файле `dps.xml`. Параметры модуля подразделяются на следующие группы:

- параметры узлов, на которых расположены экземпляры лингвистического процессора;
- параметры узла, на котором расположен экземпляры плагина для преобразования web-страниц в DOM-документы;
- прочие параметры.

Параметры узлов, на которых расположены экземпляры лингвистического процессора, определяются отдельными секциями `<endpoint>`, объединёнными в список `<endpoints>`. Допустимыми вложенными тэгами для секции `<endpoint>` являются:

Параметр	Описание
<host>	– адрес узла, на котором расположен экземпляр лингвистического процессора
<port>	– порт, используемый лингвистическим процессором
<knowledge>	– идентификатор предметной области, на работу с которой настроен экземпляр лингвистического процессора
<power>	– количество запросов, параллельно выполняемых экземпляром лингвистического процессора, размещённым на узле. Как правило, используется количество процессоров, умноженное на 2-3
<threshold>	– порог пополнения очереди DPS. Обычно выставляется в $0.7 * power$

При нескольких доступных узлах задания распределяются модулем DPS таким образом, чтобы загрузка узлов была равномерной (с учётом значений параметров <power> и <threshold>).

Параметры узла, на котором расположен экземпляр плагина для преобразования web-страниц в DOM-документы, определяются секцией <domendpoint>. Допустимыми вложенными тэгами для секции <domendpoint> являются:

Параметр	Описание
<power>	– количество запросов, параллельно выполняемых экземпляром плагина, размещённым на узле
<threshold>	– порог пополнения очереди DPS, рекомендуемое значение: $0.7 * power$ (проверено экспериментально)

Прочие параметры модуля DPS определяются следующими тэгами:

Параметр	Описание
<encoding>	– ожидаемая кодировка текстовых файлов, являющихся входными данными операции plain
<delay>	– минимальный интервал между асинхронными запросами к лингвистическому процессору
<xmlls>	– генерировать XML с результатами разбора
<xmllsnear>	– помещать xml рядом с оригинальным файлом
<inputregex>	– регулярное выражение для отбора файлов с входными данными в каталоге файловой системы

- <failonerror> – завершать процесс обработки при первой ошибке разбора
- <domtaskproxy> – адрес узла, на котором размещён плагин для преобразования web-страниц в DOM-документы

Примером конфигурационного файла dps.xml является следующий XML-документ:

```
<?xml version="1.0" encoding="UTF-8"?>
<settings>
  <domtaskproxy>/Users/proxy/proxy.rb</domtaskproxy>
  <encoding>utf-8</encoding>
  <delay>750</delay>
  <failonerror>>false</failonerror>
  <dump>>false</dump>
  <xmles>true</xmles>
  <xmlesnear>true</xmlesnear>
  <inputregex>.*\.htm</inputregex>
  <endpoints>
    <endpoint enabled='false'>
      <host>server</host>
      <port>8090</port>
      <knowledge>Привет</knowledge>
      <power>3</power>
      <threshold>2</threshold>
    </endpoint>
    <endpoint enabled='false'>
      <host>localhost</host>
      <port>8090</port>
      <knowledge>Мир</knowledge>
      <power>3</power>
      <threshold>2</threshold>
    </endpoint>
  </endpoints>
  <domendpoint>
    <power>3</power>
    <threshold>2</threshold>
  </domendpoint>
</settings>
```

ПРИЛОЖЕНИЕ 2. ОПИСАНИЕ МОДУЛЯ ТЕСТИРОВАНИЯ ПРАВИЛ ПРЕДМЕТНОЙ ОБЛАСТИ TFW

Общие сведения

Модуль TFW представляет собой программное обеспечение, запускаемое из командной строки и предназначенное для автоматизированного тестирования функциональных возможностей контрольной версии лингвистического процессора и правил предметной области «АРИОН-Лингво» на основе сравнения с возможностями эталонной (стабильной) версии.

Основным методом тестирования является регрессионное тестирование – анализ различий в протоколах обработки входных данных, что позволяет судить о правильности не только конечного результата, но и основных стадий процесса. В процессе анализа протоколов контролируется наличие маркеров ошибок, соответствие записей протокола ожидаемым стадиям процесса, рассчитываются и сопоставляются с пороговыми значениями агрегированные характеристики.

Программный код и конфигурация модуля TFW представляет собой совокупность исполняемых файлов среды `bash shell` и скриптов на универсальном языке программирования Perl 5. Для корректной работы модуля необходимо задать значения нескольких переменных окружения.

Общая схема работы

Общая схема работы модуля TFW включает следующие основные этапы:

1. Для каждой предметной области запускаются контрольная и эталонная версии лингвистического процессора и правил предметной области с соответствующей конфигурацией.
2. Каждый тестовый блок входных данных (тестовый файл) передаётся для обработки обеим версиям лингвистического процессора. Протоколы обработки сохраняются в отдельных файлах, названия которых включают идентификатор предметной области и название тестового файла. Каталог размещения протоколов определяется версией лингвистического процессора (контрольной или эталонной).
3. На основе анализа сходства и различия протоколов обработки каждого тестового файла контрольной и эталонной версиями лингвистического процессора формируются как описания конкретных ошибок, так и общие статистические сведения о результатах тестирования.

Параметры конфигурации

Запуск модуля TFW осуществляется в командной строке или командном файле инструкцией следующего вида:

```
tr.sh [аргументы]
```

Здесь [аргументы] – параметры модуля, определяемые в конфигурационном файле или командной строке в виде пар <параметр>=<значение>.

К основным параметрам конфигурации модуля относятся:

Параметр	Описание
BINDIR	– путь к каталогу с исполняемыми файлами контрольной версии лингвистического процессора
DPSDIR	– путь к каталогу с исполняемыми файлами модуля DPS (см. Приложение 1)
SUBJDIR	– путь к каталогу с конфигурациями предметных областей и тестовыми файлами.
REPORT_PATH	– путь к каталогу, в котором будет формироваться файл отчета <code>allresults.txt</code> . По умолчанию это текущий каталог

К параметрам регрессионного тестирования относятся:

Параметр	Описание
EBINDIR	– путь к каталогу с исполняемыми файлами стабильной версии лингвистического процессора и правил
CRTCLOGS	– значение «Y» обеспечивает сохранение протоколов обработки входных данных стабильной версией в каталоге входных данных
ZAPINKA	– значение «Y» обеспечивает ожидание нажатия клавиши <Enter> после обработки очередного блока входных данных (тестового файла)
SHOWDPSLOG	– значение «Y» обеспечивает вывод протоколов модуля DPS в стандартном потоке ввода-вывода.

Результаты работы модуля

В результате работы модуля TFW на диске формируется протокол регрессионного анализа `allresults.txt`, в который включаются обнаруженные различия и их интерпретация. Файл `allresults.txt` представляет собой текстовый файл, каждая строка которого содержит следующий набор полей, разделённых символами табуляции:

Номер поля	Содержимое поля
1	идентификатор предметной области
2	путь и название тестового файла
3	идентификатор теста (вид проверки)
4	код и описание обнаруженной ошибки

5 информация, специфичная для указанного кода ошибки

Таким образом, `allresults.txt` содержит непосредственные результаты тестирования, которые имеют плоскую структуру.

Для удобства анализа и обобщения информации после окончания всех тестов создается структурированный файл `allresults.xml`. В нем представлена та же информация, что и в `allresults.txt` с добавлением секций, в которых производится группировка обнаруженных различий по следующим основаниям:

- категории ошибок;
- источники ошибок (предметная область, текст, вид проверки);
- типы объектов, для которых обнаружены различия.

Для каждого поля, по которому производится группировка, рассчитываются соответствующие агрегированные показатели, характеризующие абсолютные и относительные уровни ошибок.

Состав модуля

В состав модуля TFW входят следующие функциональные компоненты:

Название файла	Описание
<code>tr.sh</code>	– основной запускаемый скрипт на <code>bash shell</code>
<code>calltr.sh</code>	– скрипт, запускаемый без параметров, предназначенный для автоматического запуска
<code>checks.pl</code>	– библиотека функций на <code>Perl 5</code> , предоставляющая механизм выявления различий в протоколах обработки входных данных
<code>logcracker.pl</code>	– библиотека дополнительных функций к <code>checks.pl</code> , содержащая функции по извлечению описаний объектов из протоколов обработки входных данных
<code>example.pl</code>	– расширенный пример использования функций библиотеки <code>checks.pl</code>