Федеральное государственное бюджетное образовательное учреждение   
высшего образования

|  |  |
| --- | --- |
| Gerb-BMSTU_01 | «Московский государственный технический университет  им. Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана) |

ФАКУЛЬТЕТ – Информатика и управления

КАФЕДРА – Информационные системы и телекоммуникации

РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе по дисциплине

Разработка программного обеспечения

на тему

Разработка компонента чтения и парсинга файла IPTables с использованием Eclipse Xtext.

Студент группы ИУ3-73 (подпись) 26.02.2018 М.К. Гаврилов

Руководитель курсовой работы (подпись) 26.02.2018 А.М Иванов

Москва, 2018

Реферат

В работе предложена архитектура компонента чтения и парсинга файла IPTables с использованием Eclipse Xtext. Архитектура реализована с использованием объектно-ориентированного языка программирования Java и утилиты для IDE Eclipse Xtext. В целях проверки работоспособности программного обеспечения разработана методика тестирования с помощью библиотеки JUnit. Результаты проведенных тестов доказывают функциональность компонента.

Содержание

[Введение 4](#_Toc507424032)

[1 Исследовательская часть 5](#_Toc507424033)

[1.1 Выбор и обоснование способа работы с файлом конфигурации IPTables 5](#_Toc507424034)

[1.2 Обзор функциональных возможностей утилиты Xtext на разных платформах 5](#_Toc507424035)

[2 Конструкторская часть 6](#_Toc507424036)

[2.1 Техническое задание 6](#_Toc507424037)

[2.2 Выбор технических решений, удовлетворяющих интересам ЗС 6](#_Toc507424038)

[2.3 Структура проекта 6](#_Toc507424039)

[2.3.1 Плагин com.bmstu.iptables.xtext 7](#_Toc507424040)

[2.3.2 Плагин com.bmstu.iptables.xtext.ide 7](#_Toc507424041)

[2.3.3 Плагин com.bmstu.iptables.xtext.ui 7](#_Toc507424042)

[2.4 Алгоритм работы ПО 10](#_Toc507424043)

[3 Технологическая часть 11](#_Toc507424044)

[3.1 Запуск разработанного приложения 11](#_Toc507424045)

[3.2 Анализ исходного кода с помощью метрик качества 11](#_Toc507424046)

[3.3 Анализ зависимостей в коде системы 12](#_Toc507424047)

[3.4 Тестирование на корректность работы 13](#_Toc507424048)

[3.5 Реализация Junit тестов 13](#_Toc507424049)

[Приложения 15](#_Toc507424050)

[Заключение 16](#_Toc507424051)

[Список литературы 17](#_Toc507424052)

Введение

В наше время все больше и больше компаний уделяют внимание сетевому трафику.

Мониторинг сети является неотъемлемой частью любого производства. Для мониторинга сети необходимы некоторые правила, исходя из которых мы сможем выдавать либо запрещать права определенным группам пользователей. Как раз для этого и используется утилита IPTables.

Iptables — утилита командной строки, является стандартным интерфейсом управления работой межсетевого экрана (брандмауэра) Netfilter для ядер Linux, начиная с версии 2.4. С её помощью администраторы создают и изменяют правила, управляющие фильтрацией и перенаправлением пакетов. Для работы с семейством протоколов IPv6 существует отдельная версия утилиты — Ip6tables. Для использования утилиты Iptables требуются привилегии суперпользователя (root). [1]

Более подробно можно рассмотреть суть данной утилиты на рисунке 1.

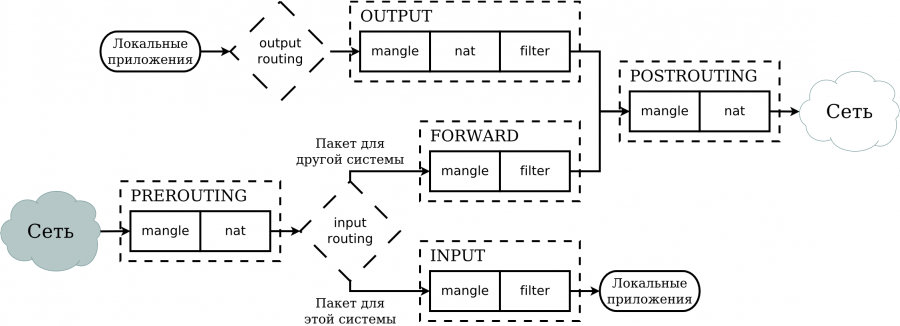


Рисунок 1 - Диаграмма прохождения таблиц и цепочек

1. Исследовательская часть

Настоящий раздел является теоретическим и включает:

* Выбор способа работы с файлом конфигурации IPTables;
* Обзор функциональных возможностей Xtext.
  1. Выбор и обоснование способа работы с файлом конфигурации IPTables

Для работы с файлом конфигурации IPTables был выбран компонент IDE Eclipse Xtext.

Xtext - это основа для разработки языков программирования и доменных языков. С помощью Xtext можно определить свой язык с использованием мощного языка грамматики. В результате можно получаете полную инфраструктуру, включая парсер, компоновщик, typechecker, компилятор, а также поддержку редактирования в Eclipse, любого редактора, который поддерживает протокол Language Server. [2]

* 1. Обзор функциональных возможностей утилиты Xtext на разных платформах

На рисунке 2 показаны функциональные возможности Xtext на различных платформах.

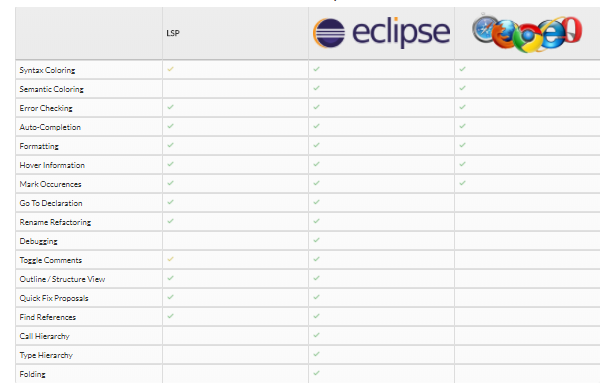


Рисунок 2 - Возможности редактора на различных платформах

Исходя из данной таблицы ми видим, что такие функции как подсветка синтаксиса, семантическая подсветка (предложение выбора дальнейших действий), проверка на наличие ошибок, авто-заполнение полей, проверка соответствию формату и другие присутствуют во всех средах разработки и различных платформах.

Они же, в свою очередь, и являются основными функциональными возможностями данной утилиты Xtext.

1. Конструкторская часть
   1. Техническое задание

Изучить соответствующие системы, спроектировать интерфейс компонента, реализовать компонент, спроектировать JUnit тесты, провести тестирование, описать требования, конструкцию, особенности сборки проекта в документации.

* 1. Выбор технических решений, удовлетворяющих интересам ЗС

В таблице ниже представлены результаты выявления и начального анализа заинтересованных сторон (ЗС) и их интересов по отношению к системе. [3]

|  |  |
| --- | --- |
| Интересы заинтересованных сторон | Технические решения |
| Подсветка синтаксиса, удовлетворяющего, требованиям IPTables в редакторе. | Утилита Xtext позволяет выполнить данное требование. |
| Обнаружение и уведомление о возникающих ошибках в файле конфигураций IPTables. | Созданный редактор, благодаря утилите Xtext поддерживает данную функцию. |
| Предложение последующих возможных вариантов данных для работы с файлом конфигурации IPTables. | Xtext позволяет выполнить и этот пункт условий. |
| Возможность парсинга и проверки на валидность уже готовых сборок конфигурации IPTables. | С помощью Junit тестов, протестирован данный сценарий развития событый. |

Таблица 1 – Заинтересованные стороны и их интересы по отношению к системе

* 1. Структура проекта

Разберем работу написанной системы, для этого опишем состав каждого плагина, из которых состоит наша система:

|  |  |
| --- | --- |
| Плагин | Содержимое |
| com.bmstu.iptables.xtext | содержит определение грамматики и все языковые компоненты (парсер, лексер, компоновщик, валидатор и т. д.) |
| com.bmstu.iptables.xtext.ide | содержит независимые от платформы функциональные возможности IDE (например, службы поддержки контента) |
| com.bmstu.iptables.xtext.ui | содержит редактор Eclipse и другие связанные с Workbench функциональные возможности |

Таблица 2 – Разработанные плагины и их описание

* + 1. Плагин com.bmstu.iptables.xtext

Данный плагин содержит 4 пакета:

* com.bmstu.iptables.xtext;
* com.bmstu.iptables.xtext.generator;
* com.bmstu.iptables.xtext.scoping;
* com.bmstu.iptables.xtext.validation.

Пакет com.bmstu.iptables.xtext содержит файл Language.xtext, который заключает в себе описание грамматики языка и другие автоматически сгенерированные файлы, необходимые для корректной работы редактора.

Файл Language.xtext содержит в себе грамматику, необходимую для правильной работоспособности редактора файла конфигурации IPTables.

Листинг данного файла находится в приложении 1.

В данном файле описана типичная последовательность и структура файла конфигураций IPTables. Как мы видим начинается он всегда с «шапки» со слов цепочки (“Chain”) далее идут уже параметры каждого пакета.

Пакет com.bmstu.iptables.xtext.generator содержит файл LanguageGenerator.xtend, который генерирует код из файлов модели при сохранении.

Пакет com.bmstu.iptables.xtext.scoping содержит файл LanguageScopeProvider.xtend, котором содержится описание настраиваемой области.

* + 1. Плагин com.bmstu.iptables.xtext.ide

Данный плагин содержит 1 пакет:

* com.bmstu.iptables.xtext.ide

Пакет com.bmstu.iptables.xtext.ide содержит 2 класса:

* LanguageIdeModule.xtend;
* LanguageIdeSetup.xtend.

|  |  |
| --- | --- |
| Класс | Описание |
| LanguageIdeModule.xtend | Используется для регистрации компонентов IDE |
| LanguageIdeSetup.xtend | Используется для инициализации для запуска языков Xtext в качестве языковых серверов. |

Таблица 3 – Классы пакета com.bmstu.iptables.xtext.ide и их описание

* + 1. Плагин com.bmstu.iptables.xtext.ui

Данный плагин содержит 5 пакетов:

* com.bmstu.iptables.xtext.ui;
* com.bmstu.iptables.xtext.ui.contentassist;
* com.bmstu.iptables.xtext.ui.labeling;
* com.bmstu.iptables.xtext.ui.outline;
* com.bmstu.iptables.xtext.ui.quickfix.

Пакет com.bmstu.iptables.xtext.ui содержит 1 класс:

* LanguageUiModule.xtend.

|  |  |
| --- | --- |
| Класс | Описание |
| LanguageUiModule.xtend | Поддержка инициализации языка Xtext в качестве языкового сервера |

Таблица 4 – Классы пакета com.bmstu.iptables.xtext.ui и их описание

Пакет com.bmstu.iptables.xtext.ui.contentassist содержит 1 класс:

* LanguageProposalProvider.xtend.

|  |  |
| --- | --- |
| Класс | Описание |
| LanguageProposalProvider.xtend | Поставщик языковых предложений |

Таблица 5 – Классы пакета com.bmstu.iptables.xtext.ui.contentassist и их описание

Пакет com.bmstu.iptables.xtext.ui.labeling содержит 2 класса:

* LanguageDescriptionLabelProvider.xtend;
* LanguageLabelProvider.xtend.

|  |  |
| --- | --- |
| Класс | Описание |
| LanguageDescriptionLabelProvider.xtend | Предоставляет метки для IEObjectDescriptions и IResourceDescriptions |
| LanguageLabelProvider.xtend | Предоставляет метки для EObjects |

Таблица 6 – Классы пакета com.bmstu.iptables.xtext.ui.labeling и их описание

Пакет com.bmstu.iptables.xtext.ui.outline содержит 1 класс:

* LanguageOutlineTreeProvider.xtend.

|  |  |
| --- | --- |
| Класс | Описание |
| LanguageOutlineTreeProvider.xtend | Настройка структуры выделения по умолчанию |

Таблица 7 – Классы пакета com.bmstu.iptables.xtext.ui.outline и их описание

Пакет com.bmstu.iptables.xtext.ui.quickfix содержит 1 класс:

* LanguageOutlineTreeProvider.xtend.

|  |  |
| --- | --- |
| Класс | Описание |
| LanguageQuickfixProvider.xtend | Пользовательские быстрые исправления |

Таблица 8 – Классы пакета com.bmstu.iptables.xtext.ui.quickfix и их описание

* 1. Алгоритм работы ПО

Окончательный алгоритм работы разработанного ПО следующий:  
1) Написание или изменение грамматики языка;  
2) Запуск генерации языковых артефактов;  
3) Запуск сгенерированного плагина в Eclipse;   
4) Создание нового проекта;  
5) Создания файла с расширением \*.iptbls в созданном проекте;  
6) Использование редактора.

1. Технологическая часть
   1. Запуск разработанного приложения

Исходный код проекта доступен в репозитории GitHub [4].

* 1. Анализ исходного кода с помощью метрик качества

На рисунке 3 показано соотношение пакетов проекта по их размеру. Видно, что самый большой размер имеет пакет language.



Рисунок 3 – Соотношение пакетов по размеру

Далее на рисунке 4 отображен список всех метрик по разделам. Всего имеется четыре раздела:

* метрики количества (Count);
* метрики сложности (Complexity);
* метрики Роберта Мартина (Robert C. Martin);
* метрики Чидамбера-Кемерера (Chidamber & Kermerer).
* Первый раздел с метриками количества (Count) содержит следующие метрики:
* количество классов верхнего уровня (Unit);
* среднее число внутренних классов на класс (Classes / Class);
* среднее число методов в классе (Methods / Class);
* среднее число полей в классе (Fields / Class);
* число строчек кода (ELOC);
* число строчек кода на модуль (ELOC / Unit).

Второй раздел с метриками сложности (Complexity) содержит всего три различных метрики:

* средняя циклическая сложность (CC);
* метрика Fat (Fat);
* средняя зависимость компонентов между модулями (ACD - Unit).

Третий раздел с метриками Роберта Мартина содержит следующие метрики:

* нормализованное расстояние от основной последовательности (D);
* абстрактность (A);
* нестабильность (I);
* число афферентных соединений (Ca);
* число эфферентных соединений (Ce).

Четвертый раздел с метриками Чидамбера-Кемерера содержит следующие метрики:

* средняя длина метода на класс (WMC);
* средняя глубина наследования (DIT);
* среднее количество классов-наследников (NOC);
* среднее число соединений класса (CBO);
* среднее число методов, которые потенциально могут быть выполнены в ответ на сообщение, полученное объектом этого класса (RFC);
* отсутствие единства методов (LCOM).

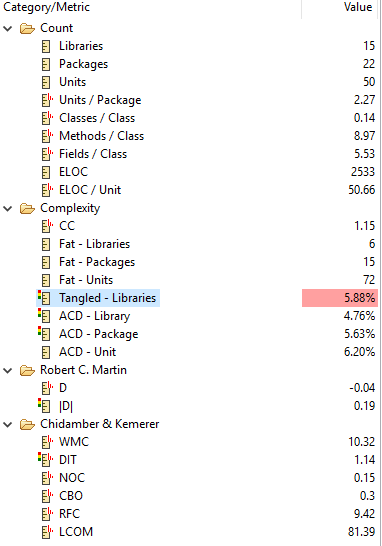


Рисунок 4 – Значения метрик

* 1. Анализ зависимостей в коде системы

Рисунок 5 описывает зависимости между пакетами разработанного программного обеспечения:

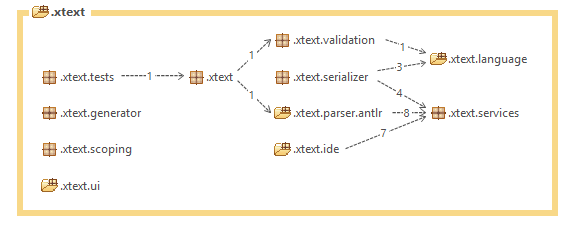


Рисунок 5 – Зависимости между пакетами

* 1. Тестирование на корректность работы

Работоспособность разработанного программного обеспечения проверялась на тесте, задачей которого является проверить работоспособность парсера.

На рисунке 6 показан редактор файла конфигураций IPTable с корректными данными:

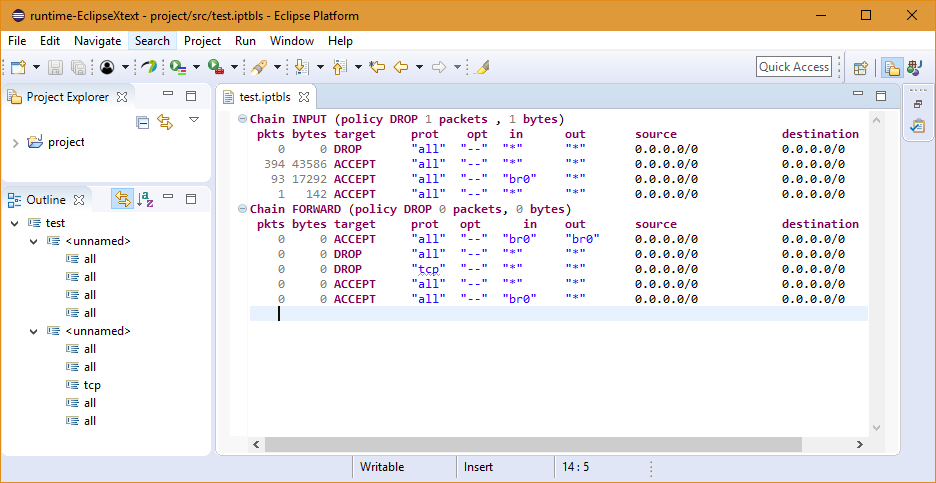
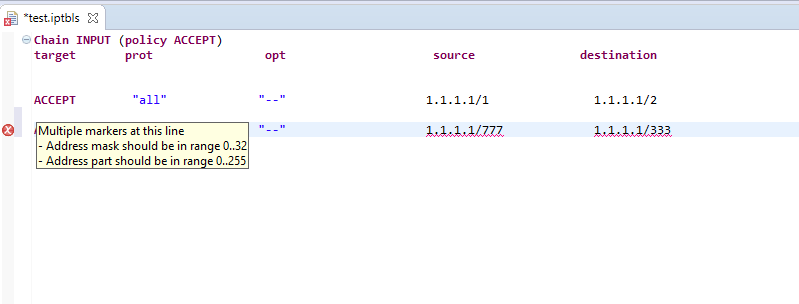


Рисунок 6– Работоспособность редактора с корректными данными

На рисунке 7 показан редактор, уведомляющий об ошибке ввода IP-адреса и маски.

  
**Рисунок 7 – Работоспособность редактора с некорректными данными**

* 1. Реализация Junit тестов

Для последующего тестирования разработанного программного обеспечения была поставлена задача написания JUnit тестов.

.

В ходе выполнения курсовой работы было написан 1 тест, проверяющий следующий функционал ПО:

Работоспособность парсера содержимого файла конфигураций IPTables и распознавание грамматики языка.

Перед тестирующими методами используется аннотация Test.

Результаты тестирования JUnit приведены на рисунке 8

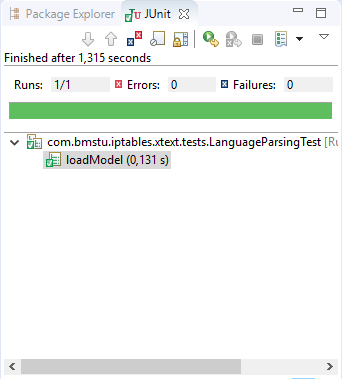


Рисунок 8 – Результат работы тестов для проверки на корректность работы

Приложения

Приложение 1 – Листинг файла Language.xtext

**grammar** com.bmstu.iptables.xtext.Language **with** org.eclipse.xtext.common.Terminals

**generate** language "http://www.bmstu.com/iptables/xtext/Language"

Model:

chains+=Chain\*;

Chain:

'Chain' chainType=ChainType '(''policy' policyType=PolicyType counter1=INT 'packets'','counter2=INT 'bytes'')'

'pkts' 'bytes' 'target' 'prot' 'opt' 'in' 'out' 'source' 'destination'

parts+=ChainPart\*;

**enum** ChainType:

INPUT='INPUT'

| OUTPUT = 'OUTPUT'

| FORWARD = 'FORWARD'

| PREROUTING = 'PREROUTING'

| POSTROUTING = 'POSTROUTING';

**enum** PolicyType:

DROP = 'DROP'

| ACCEPT = 'ACCEPT';

ChainPart:

pkts=INT bytes=INT target=PolicyType prot=STRING opt=STRING in=STRING out=STRING source=IP\_ADDRESS destination=IP\_ADDRESS;

**terminal** IP\_ADDRESS: 'anywhere' | '!'?INT'.'INT'.'INT'.'INT'/'INT(',''!'?INT'.'INT'.'INT'.'INT'/'INT)\*;

Заключение

В ходе выполнения курсовой работы был разработан редактор файла конфигурации IPTables. Работа была выполнена с помощью технологий Xtext и JUnit в среде разработки Eclipse.

В результате выполнения проекта были решены задачи, поставленные в начале работы. Были изучены основные принципы работы необходимых технологий; разработано ПО, соответствующее требованиям ТЗ; проведены необходимые тесты работоспособности программного продукта. Результаты тестирования свидетельствуют правильное функционирование ПО.

Список литературы

[1]  Iptables | Русскоязычная документация по Ubuntu: [Электронный ресурс]. Режим доступа: http://help.ubuntu.ru/wiki/iptables

[2] Xtext - Language Engineering Made Easy! - Eclipse: [Электронный ресурс]. Режим доступа: https://eclipse.org/Xtext/index.html

[3] Xtext 15 Minutes Tutorial : [Электронный ресурс]. Режим доступа: https://www.eclipse.org/Xtext/documentation/102\_domainmodelwalkthrough.html

[4] GitHub – IPTables [Электронный ресурс]. Режим доступа: https://github.com/GavrilovMike/iptables