ISSN 2500-1752

Международный журнал информационных технологий и энергоэффективности

Том 4 Номер 1(11)

2019

Международный журнал информационных технологий и энергоэффективности
Т.4 №1(11) 2019 г.

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

СОДЕРЖАНИЕ / CONTENT

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

Певченко С.И. Применение системы моделирования Deeds в дисциплинах «Микропроцессорные системы» и «Схемотехника ЭВМ» // Международный журнал информационных технологий и энергоэффективности. – 2019. – Т.4 №1(11) с. 3-9

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.94

ПРИМЕНЕНИЕ СИСТЕМЫ МОДЕЛИРОВАНИЯ DEEDS В ДИСЦИПЛИНАХ «МИКРОПРОЦЕССОРНЫЕ СИСТЕМЫ» И «СХЕМОТЕХНИКА ЭВМ»

Певченко С.И.

ФГБОУ *ВО Национальный исследовательский институт «Московский энергетический институт», Россия, (111250, г. Москва, ул. Красноказарменная, 14), e-mail: pevserg@yandex.ru*

Статья посвящена вопросу верификации курсовых проектов студентов по дисциплине «Микропроцессорные системы». Отмечается целесообразность использования для непрофильных ВУЗов простой в установке и изучении свободно распространяемой учебной системы цифрового моделирования Deeds, состав пользовательских блоков которой дополнен автором типовыми блоками микропроцессорных систем (МПС).

Ключевые слова: учебная система моделирования Deeds, проектирование микропроцессорных систем (МПС), верификация курсовых проектов МПС.

APPLICATION OF THE MODELING SYSTEM DEEDS IN DISCIPLINES «MICROPROCESSOR SYSTEMS» AND «CIRCUIT ENGINEERING OF A COMPUTER»

Pevchenko S.I.

Federal State Educational Institution of Higher Education National Research University «Moscow Power Engineering Institute», Russia, (111250, Moscow, street Krasnokazarmennaya, 14), e-mail: pevserg@yandex.ru

The paper is devoted to an issue of verification coursework's students project on discipline «Microprocessor systems». The expediency of use for non-core higher educational institute idle time in installation and studying of freely distributed educational system of digital modeling Deeds which structure of the user blocks is complemented with the author standard blocks of microprocessor systems (MPS) is noted.

Key words: educational modeling system Deeds, designing microprocessor system (MPS), verification of microprocessor system course project.

В сравнении с профильными ВУЗами типа МИЭТ, МИЭМ, МИФИ, глубоко изучающими вопросы проектирования МПС и микропроцессоров (МП), в непрофильных ВУЗах типа МЭИ обычно ограничиваются вопросами построения МПС на базе стандартного МП типа i8080 с реализацией управляющей программы на ассемблерном языке и использовании в качестве внешних устройств типовых блоков: память, АЦП, ЦАП, контроллеры прерываний, клавиатуры и т.п.

Типовые задания курсового проектирования на кафедре ВМСС МЭИ предполагают разработку структурной и принципиальной электрических схем МПС и управляющей программы с оформлением по ЕСКД и ЕСПД соответственно. Однако верификация проектов моделированием заданием не предусматривается, что резко снижает качество защищаемых проектов. Применение систем моделирования, подобных Deeds, позволяет преодолеть этот недостаток.

Система моделирования Deeds, разработанная в Генуэзском Университете [2,4], предназначена для обучения проектированию цифровых систем и свободно распространяется разработчиками [5].

Система состоит из трех взаимосвязанных компонентов:

- Deeds-DcS (Digital Circuit Simulator) — подсистема моделирования цифровых схем. Ее основные достоинства и недостатки представлены в таблице 1.
- Deeds-FsM (Finite State Machine Simulator) — подсистема моделирования конечных автоматов.
- Deeds-McE (Micro Computer Emulator) — подсистема моделирования микро-ЭВМ семейства i8080.

Таблица 1 – Достоинства и недостатки подсистемы моделирования цифровых схем

Достоинства	**Недостатки**
Множество описаний лабораторных работ по изучению типовых узлов, автоматов и МПС на сайте разработчика системы [5] и методические пособия на кафедры ВМСС	Отсутствует иерархия вложенности пользовательских блоков произвольной глубины
Применение системы Deeds в курсах «Микропроцессорные системы», «Схемотехника ЭВМ» и «Теория автоматов»	Нельзя задать индивидуальные задержки базовых (библиотечных) элементов
Простота пользовательского интерфейса и схемного редактора	Отсутствие русифицированной версии
Поддержка четырехзначного алфавита моделирования (0, 1, X, Z), позволяющего описывать связи типа общая шина	Отсутствие библиотек микросхем промышленных серий
Поддержка создания пользовательских блоков из стандартного набора блоков системы Deeds	
Получение текстового описания схем на языке VHDL [3] из их графического описания	
Два режима моделирования: интерактивный «симулятор» (режим наглядной анимации) и временное моделирование (получение временных диаграмм)	
Реализация физического эксперимента на отладочных платах с ПЛИС при условии использования средств САПР ПЛИС фирмы Altera	

Подсистема цифрового моделирования схем поддерживает два режима моделирования работы схемы.

- Интерактивный симулятор («Анимация»).
- Временное моделирование схемы.

В режиме «анимации» воспроизводится по тактам поведение схемы. Таким образом, можно визуально отслеживать изменение цвета индикаторов сигналов, в том числе и внутренних сигналов пользовательского блока, для чего достаточно в этом режиме нажать на соответствующий блок, предварительно установив в нем контрольные точки (TEST POINT).

Результатом временного моделирования схемы являются временные диаграммы входных и выходных сигналов. Также можно видеть и внутренние сигналы пользовательского блока. При этом имеется возможность выбора конкретного списка наблюдаемых сигналов. Внутренние сигналы пользовательского блока скрыты по умолчанию.

При разработке графический представлений проектируемых схем пользователь использует схемный редактор. В качестве базовых компонентов в библиотеке подсистемы цифрового моделирования схем следует отметить несколько основных групп.

- **Входные переключатели** - группа элементов, предназначенная для подачи входных сигналов на этапе анимационного и временного моделирования. Существуют элементы с шинными выходами (bus) и скалярными.
- **Индикаторы выходных сигналов** – группа элементов, предназначенная для отслеживания состояния внутренних сигналов (контрольных точек) и выходов моделируемой схемы. Существуют элементы с шинными входами (bus) и скалярными.
- **Межкомпонентные соединения (шины и провода)** - группа элементов-соединителей: провода, шины, шинные разветвители (bus splitter) и ответвители (bus tap) для установления логических связей между компонентами схемы.

Элементный базис системы Deeds состоит из:

1) *Комбинационных логических блоков*. Они могут быть, как простыми логическими элементами (вентили И, ИЛИ и т.п.), так и более сложными (декодеры, мультиплексоры, сумматоры, АЛУ и т.д.). Задержка вентилей около 4 нс, комбинационных узлов 8-10 нс.

2) *Запоминающих логических блоков* (триггеры, регистры, счетчики, таймеры). Задержки триггеров около 8 нс, регистров 10 нс.

Триггеры подсистемы моделирования цифровых схем различаются по:

- Логике функционирования (например RS, D и JK).
- Способу приема и выдачи информации (асинхронные, синхронные, по уровню, по фронту/спаду).

Регистры подсистемы моделирования цифровых схем подразделяются на:

- Регистры с параллельным входом и выходом (P.I.P.O).
- Регистры с параллельным входом и последовательным выходом (P.I.S.O).
- Регистры с последовательным входом и параллельным выходом (S.I.P.O).
- Универсальные регистры.

Счетчики – узлы, на выходах которых получается двоичный код, определяемый числом поступивших импульсов.

Таймеры – узлы, подающие сигнал через заданный временной интервал.

3) Микросхемы памяти ПЗУ (ROM) и ОЗУ (RAM). Блоки ОЗУ делятся на синхронные и асинхронные. Задержки блоков памяти примерно равна 20 нс.

4) Устройства вывода аналогово сигнала (ЦАП).

5) Микроконтроллер (МК) DMC8, существующий в двух модификации: DMC8 Basic и DMC8 Enhanced. Основные отличия этих модификаций в системе команд (Zilog Z80 и Intel 8080), количестве входных и выходных портов, а также системе прерывания.

Подсистема моделирования микро-ЭВМ применяется для разработки и отладки программного обеспечения МПС на базе двух модификаций МК DMC8. В этой подсистеме применяется низкоуровневый язык программирования – ассемблер.

В отладчике (см. рисунок 1) можно пройти код в пошаговом режиме (Step), режиме анимации (Animate) и в реальном режиме работы программы (Run), а так же задавать скорость анимации.

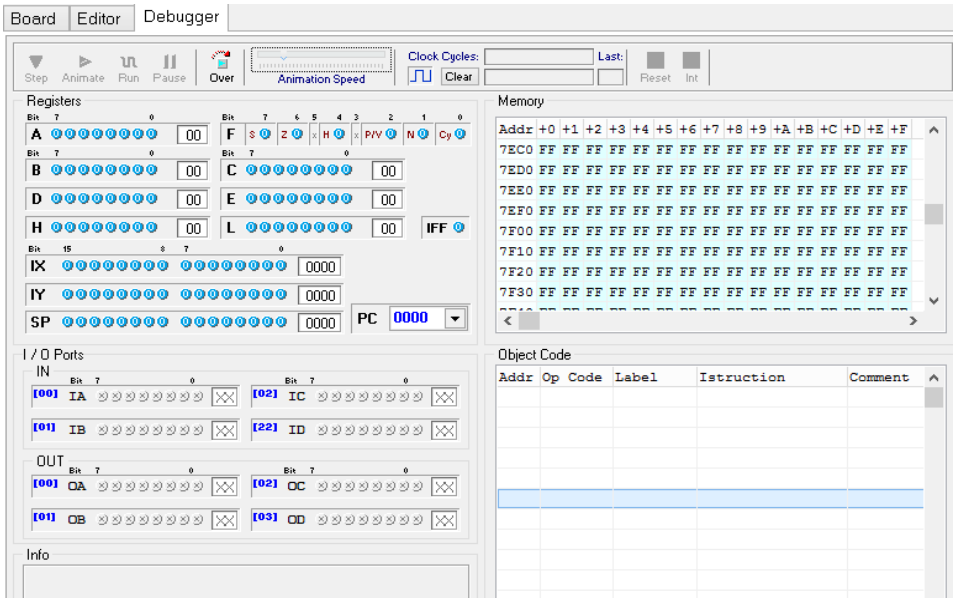

Рисунок 1 – Окно отладчика подсистемы моделирования микроэвм

Кроме того, в нем показывается содержимое регистров, ОЗУ, значение ячеек которого можно изменять в пошаговом и анимационном режимах во время отладки программы и содержимое ПЗУ, которое изменяется только при компиляции кода.

Физические эксперименты на отладочных платах.

Ранее отмечалось, что система моделирования Deeds поддерживает стыковку с САПР ПЛИС фирмы Altera Quartus II, позволяя выполнять тестирование разработанной пользователем схемы на основных отладочных платах ПЛИС этой фирмы.

Для запуска этого процесса на главной панели схемного редактора необходимо нажать на пиктограмму Test on FPGA. В открывшемся окне (см. рисунок 2) выбирается из списка название отладочной платы, после чего назначаются входные и выходные шинные и скалярные сигналы схемы на кнопки, переключатели и индикаторы выбранной отладочной платы.

Для завершения этапа конфигурирования отладочной платы запускаем процесс создания файла программирования ПЛИС (Generate Project). В результате сформируется папка с проектом, в которой будет храниться прошивка ПЛИС, которую по кабелю можно переслать из персонального компьютера в плату.

Процесс разработки МПС с использованием средств системы моделирования Deeds можно разделить на обязательную и второстепенную части. Они в свою очередь делятся на несколько этапов (см. рисунок 3).

1) *На первом этапе* разрабатывается структурная и функциональная схемы МПС в соответствии со спецификацией типового задания на курсовой проект с использованием основных блоков моделей цифровых устройств, разработанных на кафедре ВМСС.

2) *На втором этапе* разрабатывается программное обеспечения для МПС, производится его отладка в подсистеме моделирования микроэвм и ввод в МК DMC8.

3) *На третьем этапе* выполняется верификация всего проекта МПС в режиме временного моделирования с целью определения её работоспособности. Режим анимации можно использовать как для тестирования проекта, так и для наглядной демонстрации.

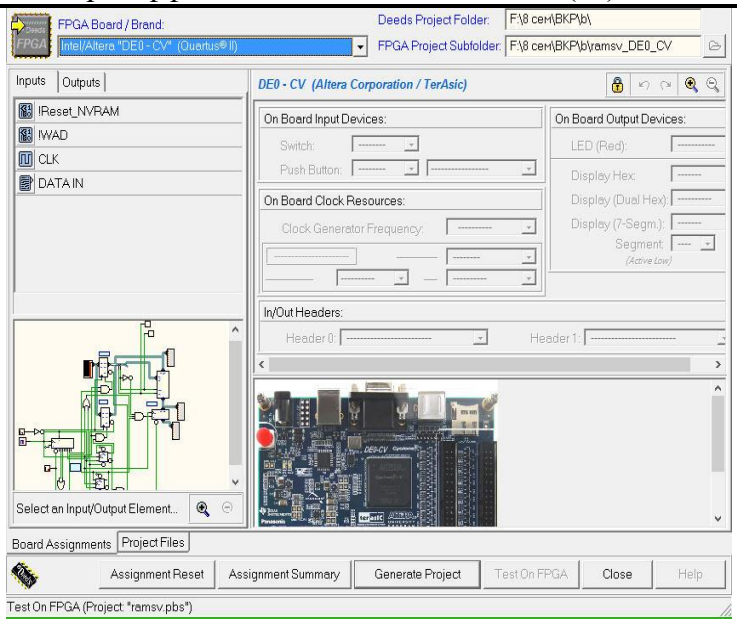

Рисунок 2 – Окно подготовки теста на отладочной плате (Test on FPGA)

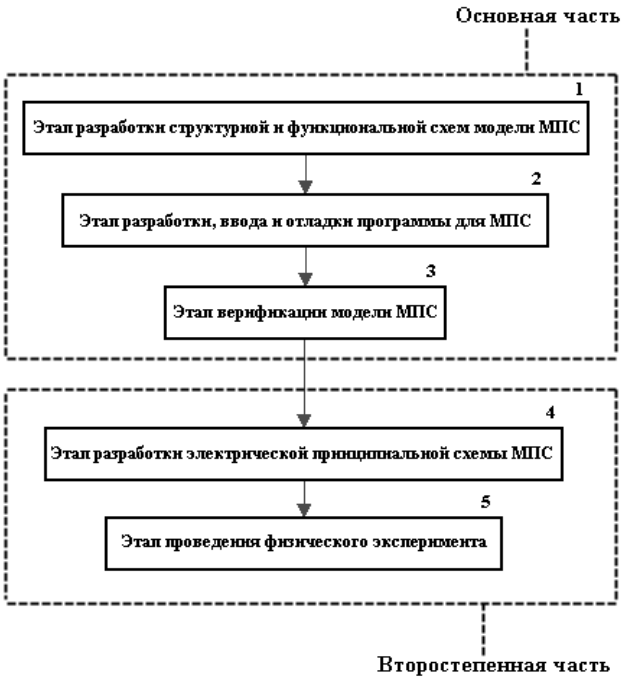

Рисунок 3 – Алгоритм процесса разработки МПС

Четвертый и пятый этапы являются не обязательными.

4) *На четвертом этапе* на основе модели МПС проектируется принципиальная схема МПС с использованием библиотеки микросхем промышленной серии.

5) *На пятом этапе* для выбранной отладочной платы ПЛИС фирмы Altera полученная прошивка загружается средствами САПР этой фирмы (Quartus II).

Пример разработанной МПС с помощью средств системы моделирования Deeds представлен на см. рисунок 4).

В ней используются блоки моделей ведущего и ведомого устройств интерфейса SPI, блоки внешней памяти, системы таймеров, ЦАП и т.д. Она может принимать информацию об объекте управления от различных датчиков, вырабатывать управляющее воздействие в

соответствии с программой управления, заложенной в МК, а также получать сигналы прерывания от различных источников, к примеру: датчика напряжения питания, клавиш клавиатуры, датчика аварийной сигнализации и т.д.

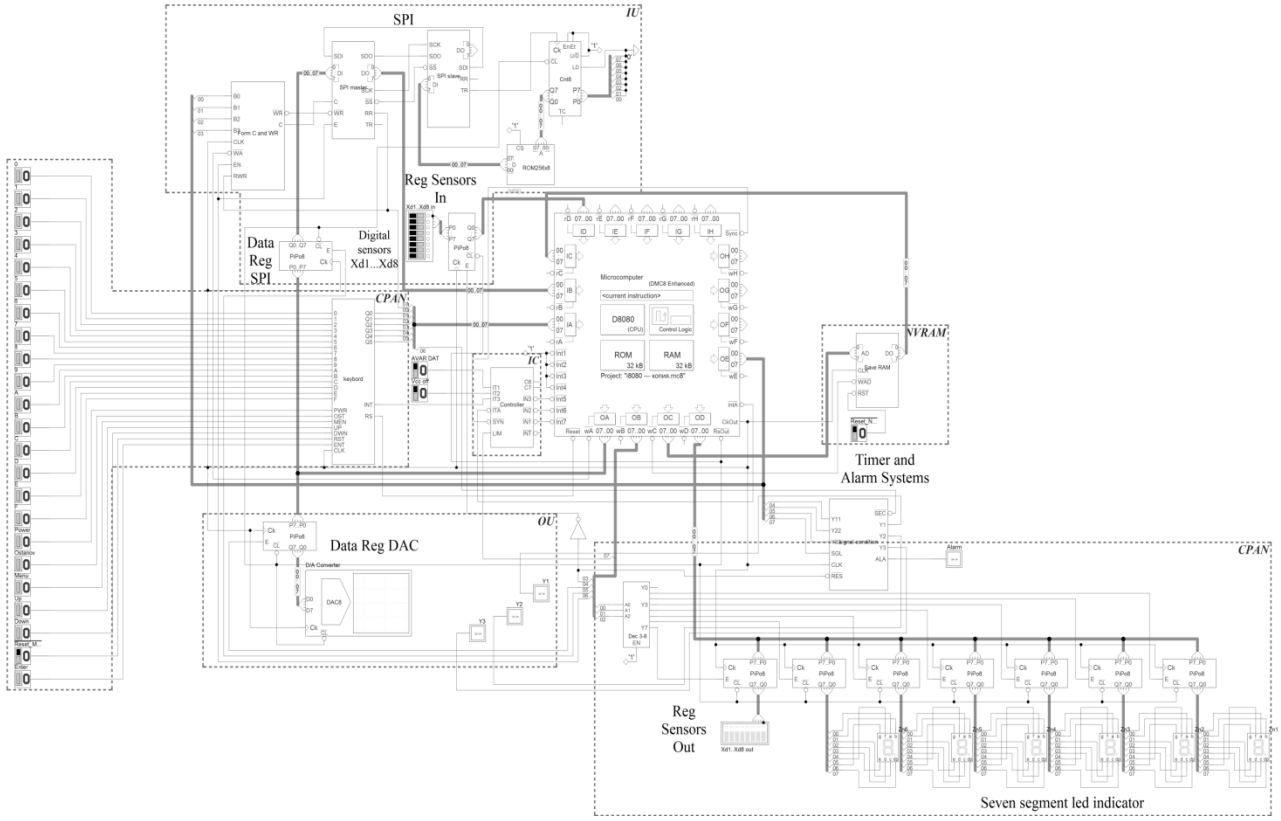

Рисунок 4 – Пример функциональной схемы МПС на базе МК DMC8 Enhanced

В качестве некоторых характеристики подобных систем можно выделить:

- Общее количество используемых блоков примерно равное 6, а суммарное число компонентов в блоках — около 135.
- Число ассемблерных инструкций (строк кода) программы управления — около 630.
- Количество используемых меток в программе — около 140.
- Объём программы управления, занимаемый в памяти ПЗУ — около 1400 байт.
- Время прогона одного цикла программы управления при временном моделировании — около 12-13 минут.

В настоящее время система моделирования Deeds используется экспериментально на кафедре ВМСС «НИУ «МЭИ» в рамках нескольких курсов:

- микропроцессорные системы — производится верификация курсовых проектов на базе МП с системой команд типа i8080;
- схемотехника ЭВМ – обучения основам цифровой схемотехники. Выполение лабораторных работ по комбинационным и последовательностным узлам.
- современные методы проектирования цифровых схем — обучение магистров, поступающих в «МЭИ» из других ВУЗов, где у них отсутствовал курс «Схемотехника ЭВМ».

Для упрощения реализации моделей курсовых проектов МПС на кафедре ВМСС созданы пользовательские блоки типовых последовательных интерфейсов (I2C, SPI, RS—232), контроллеры прерываний и клавиатуры, блок внешней памяти, система таймеров, а также библиотека интегральных микросхем промышленных серий КР1533 и КР1531 [1]. С их

помощью упрощается проектирование и верификация индивидуальных курсовых проектов МПС.

Таким образом, благодаря учебным средствам, подобным системе моделирования Deeds, реализуется возможность обучения студентов непрофильных ВУЗов основам цифровой схемотехники, моделирования и проектирования МПС.

Список литературы

1. Петровский И.И. и др. Логические ИС КР1533, КР1554. Справочник. В двух частях. — М.: Бином, 1993. — часть 1 — 254 с., часть 2 — 497 с.
2. Поляков А.К.. DEEDS — УЧЕБНАЯ СИСТЕМА МОДЕЛИРОВАНИЯ И ПРОЕКТИРОВАНИЯ ЦИФРОВОЙ АППАРАТУРЫ. Журнал. Современная Электроника №1. г. 2018, с.94-96.
3. Поляков А.К.. Языки VHDL и VERILOG в проектировании цифровой аппаратуры на ПЛИС: учебное пособие. — М.: Издательский дом МЭИ, 2012. — 220с.
4. Giuliano Donzellini, Domenico Ponta: Deeds — User Manual. : University of Genoa, 2004. — 121 с.
5. Giuliano Donzellini, Domenico Ponta. Официальный сайт программного обеспечения системы моделирования Deeds. [Электронный ресурс]. URL: https://www.digitalelectronicsdeeds.com, (дата обращения 01.04.2019)

References

1. Petrovskij I.I. and others. Logical IC KR1533, KR1554. Reference book. In two part. – M.: Binom, 1993 – part 1 – 254 p., part 2 – 497 c. (in Russia).
2. Poliakov A.K., Educational modeling and designing system of a digital equipment. Journal. Modern Electronics №1., г.2018. с.94-96. (in Russia).
3. Poliakov A.K. VHDL and VERILOG languages in a design of digital devices on the FPGA: training manual. – M.: Publish. House MPEI, 2012. – 220p. (in Russian)
4. Giuliano Donzellini, Domenico Ponta: Deeds — User Manual. : University of Genoa, 2004. — 121 с.
5. Giuliano Donzellini, Domenico Ponta. Official site of the software of the modeling system Deeds. [Electronic resource]. URL: https://www.digitalelectronicsdeeds.com, (access date 01.04.2019).

Балашов О.В., Лосева В.А. Извлечение знаний для систем поддержки принятия решений // Международный журнал информационных технологий и энергоэффективности. – 2019. – Т.4 №1(11) с. 10-15

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 681.3.06

ИЗВЛЕЧЕНИЕ ЗНАНИЙ ДЛЯ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

Балашов О.В., Лосева В.А.

Смоленский филиал АО «Радиозавод», Россия, (214027, г. Смоленск, улица Котовского,2), e-mail: smradio@mail.ru

Рассматриваются подход к автоматической кластеризации /классификации объектов по данным обучающей выборки с применением современных инструментальных средств. Результат может быть полезен при проектировании систем поддержки принятия решений.

Ключевые слова: решение, кластеризация, выбор, классификация, обучающая выборка, лингвистическое описание.

EXTRACTION OF KNOWLEDGE FOR DECISION SUPPORT SYSTEMS

Balashov O.V., Loseva V.A.

Smolensk branch of joint-stock company "Radio factory", Russia, (214027, Smolensk, street Kotovskogo, 2), e-mail: smradio@mail.ru

The approach to automatic clusterization (classification) of objects according to learning sampling with application of modern tools is considered. The result can be useful at decision support systems.

Key words: decision, clusterization, choice, classification, training sample, linguistic description.

Качество функционирования системы поддержки принятия решений (СППР) существенно зависит от содержимого её базы знаний. Как известно, существуют две основные группы методов получения знаний: прямые (интервью, изучение литературы и др.) и косвенные (анализ обучающего множества примеров, наблюдения за экспертом и др.) [1, 2]. Проведённые исследования показали, что при принятии решений в условиях неопределённости большую предпочтительность имеют методы второй группы.

В данной статье рассматривается задача автоматической кластеризации по примерам обучающей выборки с выдачей результата в виде совокупности продукционных правил вида «если – то». Решение задачи проводится с использованием инструментальных средств SPSS 13.0 [3] и See5/C5.0 [4].

Постановка задачи.

Имеются массив экспериментальных данных, представленный «примерами» в виде векторов $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$, $i = 1,2,\ldots N$, где x_{ij} – некоторые числа ($j = 1,2,\ldots n$), отражающие значения количественных признаков $x_1 \div x_n$, N – общий объем выборки (количество обучающих примеров).

Предполагается, что представленные примеры отражают некоторое, априори неизвестное число m типов различимых объектов, например, различных способов выполнения работ, типов летательных аппаратов, сортов фруктов и т.п.

Требуется: по данным обучающей выборки провести автоматическую кластеризацию представленных примеров по типам объектов, определить число таких типов (кластеров), выделить наиболее информативное подмножество признаков кластеризации, сформулировать решение в виде совокупности отмеченных выше продукционных правил, т.е. в лингвистической форме – для облегчения дальнейшей ручной или полуавтоматической классификации объектов в системах СППР.

Известные методы решения задачи кластеризации. Существует большое количество различных методов решения задачи кластеризации (см., в частности, книги [1, 2, 4-9]), однако в большинстве из них количество кластеров априори задается пользователем, исходя из каких-либо содержательных представлений о характере будущего решения. Практически неизвестны методы, в которых бы, наряду с решением задачи кластеризации проводилась оценка значимости признаков. «Ручные» вычисления по данным методам пригодны лишь для задач небольшой размерности – с числом примеров не более 20, при 2÷3 признаках классификации.

В исследуемой задаче, как число примеров, так и число признаков достаточно велико, что требует привлечения того или иного инструментального (программного) средства, реализующего те или иные алгоритмы кластеризации.

Выбор инструментальных средств. Поскольку существуют инструментальные средства (программы, программные системы), позволяющие решать подобные задачи с помощью персональных компьютеров, метод решения поставленной задачи целиком зависит от выбранного инструментального средства и его возможностей, при этом пользователь математическими деталями используемых алгоритмов может и не интересоваться (эти алгоритмы, как отмечалось, достаточно подробно описаны, например, в монографиях [5, 6]).

В качестве инструментальных средств для решения поставленной задачи в данном случае выбраны:

1) пакет для статистических вычислений SPSS, 13-я версия;

2) программа See5 (версия 1.20a).

Такой выбор поясняется не только широкими возможностями указанных программ, но и тем, что они и правила их использования достаточно подробно описаны в отечественной литературе (см. [3, 4]).

Решение задачи. Предлагаемый подход продемонстрируем на следующем иллюстративном примере.

Пусть имеются объекты двух типов (еще раз оговариваем, что это число предполагается неизвестным), каждый из которых характеризуется двумя числовыми признаками, а соответствующие объектам примеры отображены в таблице 1. Данные подвергнуты рандомизации, т.е. примеры перемешаны случайным образом; в условиях примера – для контроля – принадлежности объекта к тому или иному классу приведены в крайне правом столбце матрицы (они были известны экспериментатору, но неизвестны программе).

Этап 1. Подготовка исходных данных.

Приведенные в таблице исходные данные (10 примеров – т.е. 10 пар значений признаков x_{i1}, x_{i2}) были загружены в таблицу программы SPSS для проведения кластеризации и выявления наиболее информативных признаков.

Этап 2. Выявление числа кластеров и наиболее информативных признаков. После загрузки данных в среду программы SPSS 13.0 дальнейшие исследования базировались на возможности этой программы решать задачу кластеризации несколькими методами, из которых наибольший интерес представляют так называемый метод двухступенчатой кластеризации (TwoStep Cluster). Данный метод, реализованный в системе SPSS 13.0,

позволяет не только автоматически определять оптимальное число кластеров в наборе данных, но и выделять наиболее информативные (с точки зрения задачи кластеризации) признаки.

Таблица 1 – Примеры обучающей выборки

№№ примеров	Признаки		№ класса
	x_1	x_2	
1	9.872	12.406	1
2	11.089	10.268	1
3	-10.19	21.911	2
4	-11.663	21.068	2
5	9.886	10.167	1
6	9.102	9.207	1
7	11.367	10.591	1
8	-8.506	21.161	2
9	-10.006	21.394	2
10	-10.166	20.29	2

С использованием этого метода для исследуемой выборки данных были получены следующие результаты, отраженные в файле отчета программы, фрагменты которого приведены ниже (таблицы 2 и 3).

Двухэтапный кластерный анализ

Таблица 2 – Распределение по кластерам

Распределение по кластерам		N	% объединенных	% от итога
Кластер	1	5	50,0%	50,0%
	2	5	50,0%	50,0%
	Объединенный	10	100,0%	100,0%
Итог		10		100,0%

Таблица 3 – Профили кластеров

Профили кластеров		**Центроиды**			
		x1		x2	
		Среднее	Стд. отклонение	Среднее	Стд. отклонение
Кластер	1	-10,1062	1,11858	21,1648	,58822
	2	10,2632	,94128	10,5278	1,16981
	Объединенный	,0785	10,77977	15,8463	5,67374

Заметим, что программа выдает также информацию об отнесении каждого из примеров обучающей выборки к тому или иному кластеру (классу). Эта информация будет использована при применении второй из рассматриваемых программ.

Как видно, программа правильно выделила два класса (кластера), более того, из её выходных данных следует, что все примеры были классифицированы правильно, а оба признака оказались значимыми (с вероятностью 0,95).

Вторая из таблиц отчёта содержит статистическую информацию о центрах кластеров.

Этап 3. Лингвистическое описание классов. Исследование на данном этапе проводилось с помощью программы See5 [4], которая позволяет по данным экспериментальной выборки (а

также по выявленным для каждого примера номера класса) формировать продукционные правила для лингвистической классификации объектов. Предварительно были подготовлены 2 текстовых файла – с имеющимися данными и именами переменных (файлы **Кластер.names** и **Кластер.data**).

Файл **Кластер.names**

```
class.
class: 1,2.
x1: continuous.
x2: continuous.
```

Файл **Кластер.data**

```
1,9.872,12.406
1,11.089,10.268
2,-10.19,21.911
2,-11.663,21.068
1,9.886,10.167
1,9.102,9.207
1,11.367,10.591
2,-8.506,21.161
2,-10.006,21.394
2,-10.166,20.29
```

В файле **Кластер.data** первые элементы каждой строки отражают принадлежность объекта (примера обучающей выборки) к тому или иному классу, определенному программой SPSS.

Результаты использования программы See5 (отражаемые протоколом в файле **Кластер.out**) приведены ниже.

```
See5 [Release 1.20a]    Tue Sep 12 17:51:27 2006

   Options:
      Rule-based classifiers

Class specified by attribute `class'

Read 10 cases (3 attributes) from Кластер.data

Rules:

Rule 1: (5, lift 1.7)
      x1 > -8.506
      ->  class 1  [0.857]

Rule 2: (5, lift 1.7)
      x1 <= -8.506
      ->  class 2  [0.857]

Default class: 1
```

```
Evaluation on training data (10 cases):

    Rules
----------------
 No     Errors

  2    0( 0.0%)   <<


 (a)   (b)    <-classified as
----  ----
   5          (a): class 1
         5    (b): class 2
```

Интерпретация приведенных результатов такова: всего исследовано 10 случаев, при этом выявлено 2 продукционных правила типа «если-то». Ошибки в классификации отсутствуют. Объединяя правила, можно дать их лингвистическую интерпретацию в виде одного правила:

П: если $x_1 \leq -8.506$, то объект относится к классу 2, иначе – к классу 1.

Отметим, что программа «определила» степень уверенности в справедливости классификации по приведенным правилам 0,857. Небезынтересно заметить, что в данном случае информационно значимым для классификации оказался только один показатель – x_1.

Нетрудно проверить (см. таблицу 1), что в условиях приведенного примера задача выявления продукционных правил решена безошибочно.

Таким образом, автоматически сформулированы продукционные правила, позволяющие по натуральным значениям информативных признаков относить предъявляемый объект к тому или иному классу.

Точность полученного решения следует оценить на уровне 80÷90%, что для многих практических задач следует считать приемлемым.

Следует указать, что к получаемым с помощью предложенного подхода результатам следует относиться с известной долей осторожности (как, впрочем, ко всем статистическим выводам, сделанным на основе только экспериментальных данных), проверяя их, по возможности, другими подходами.

Список литературы

1. Лбов Г. С., Старцева Н. Г. Логические решающие функции и вопросы статистической устойчивости решений. – Новосибирск, Изд-во Ин-та математики, 1999. – 212 с.
2. Загоруйко Н. Г. Прикладные методы анализа данных и знаний. Новосибирск, Изд-во Ин-та математики, 1999. – 270 с.
3. Бююль А., Цёфель П. SPSS: искусство обработки информации, анализ статистических данных и восстановление скрытых закономерностей. – СПб.: ООО "ДиаСофтЮП", 2002. – 608 с.
4. Дюк В., Самойленко А. Data mining: учебный курс. – СПб.: Питер, 2001. – 368 с.
5. Осовский С. Нейронные сети для обработки информации. – М.: Финансы и статистика, 2002. – 344 с.

References

1. Lbov G. S., Startseva N.G. Logic decision functions and questions of statistical stability of decisions. - Novosibirsk, Publishing house In mathematicians, 1999. (in Russian)
2. Zagorujko N.G. Applied methods of the analysis of data and knowledge. Novosibirsk, Publishing house In mathematicians, 1999. (in Russian)
3. Buul A., Cefel P. SPSS: art of processing of the information, the analysis of statistical data and restoration of the latent laws. - SPb.: Open Company "DiaSoftUP", 2002. (in Russian).
4. Duk V., Samoilenko A. Data mining: a training course. - SPb.: Peter, 2001. (in Russian).
5. Osovsky S. Nejronnye of a network for information processing. - M: the Finance and statistics, 2002. (in Russian).

Цепелев Ю.А., Раскатова М.В. Автоматизация написания документации с использованием искусственных нейронных сетей // Международный журнал информационных технологий и энергоэффективности. – 2019. – Т.4 №1(11) с. 16-20

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 519

АВТОМАТИЗАЦИЯ НАПИСАНИЯ ДОКУМЕНТАЦИИ С ИСПОЛЬЗОВАНИЕМ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

Цепелев Ю.А., Раскатова М.В.

Федеральное государственное бюджетное образовательное учреждение высшего образования «Национальный исследовательский университет «МЭИ», Россия (111250, г.Москва, ул. Красноказарменная, д. 14); e-mail: foxsidark@yandex.ru

В статье рассмотрен способ наполнения словаря для написания документации новыми словами с использованием искусственных нейронных сетей.

Ключевые слова: искусственные нейронные сети, алгоритм, словарь.

AUTOMATING WRITING DOCUMENTATION USING ARTIFICIAL NEURAL NETWORKS

Tsepelev Y.A., Raskatova, M. V.

National Research University “Moscow Power Engineering Institute”, Russia (111250, Moscow, Krasnokazarmennaya street, 14); e-mail: foxsidark@yandex.ru

The paper describes the method of filling the dictionary for writing documentation with new words using artificial neural networks.

Keywords: artificial neural networks, algorithm, dictionary.

В современном мире огромную роль играет документооборот и разнообразная отчетность. Для её написания требуется значительные временные затраты. Как правило, многие из отчетных документов имеют примерно одинаковый шаблон, меняются лишь небольшие части этих документов. Но изменение составных частей влияет на остальной текст, следовательно, возникает необходимость в автоматизации написания таких документов поскольку обычные шаблоны недостаточно эффективны. Для решения данной задачи интересным является применение искусственных нейронных сетей. В данной статье исследуется возможность применения алгоритма, разработанного на базе искусственных нейронных сетей, для добавления слов в словарь. Выявлены недостатки и возможные пути развития представленного алгоритма.

Идея создания искусственных нейронных сетей пришла нам из биологии, так человек может легко справляться с достаточно широким пластом задач связанными с распознаванием

образов и многими другими задачами, на которые традиционные алгоритмы тратят огромное количество вычислительных ресурсов. Так желание перенести эту способность человека привело к созданию математической модели искусственных нейронных сетей.

«Нейронная сеть – это громадный распределенный параллельный процессор, состоящий из элементарных единиц обработки информации, накапливающих экспериментальные знания и предоставляющих их для последующей обработки. Нейронная сеть сходна с мозгом с двух точек зрения.

Знания поступают в нейронную сеть из окружающей среды и используются в процессе обучения.

Для накопления знаний применяются связи между нейронами, называемые синоптическими весами.»

В свою очередь искусственная нейронная сеть состоит из нейронов [1], рисунок 1.

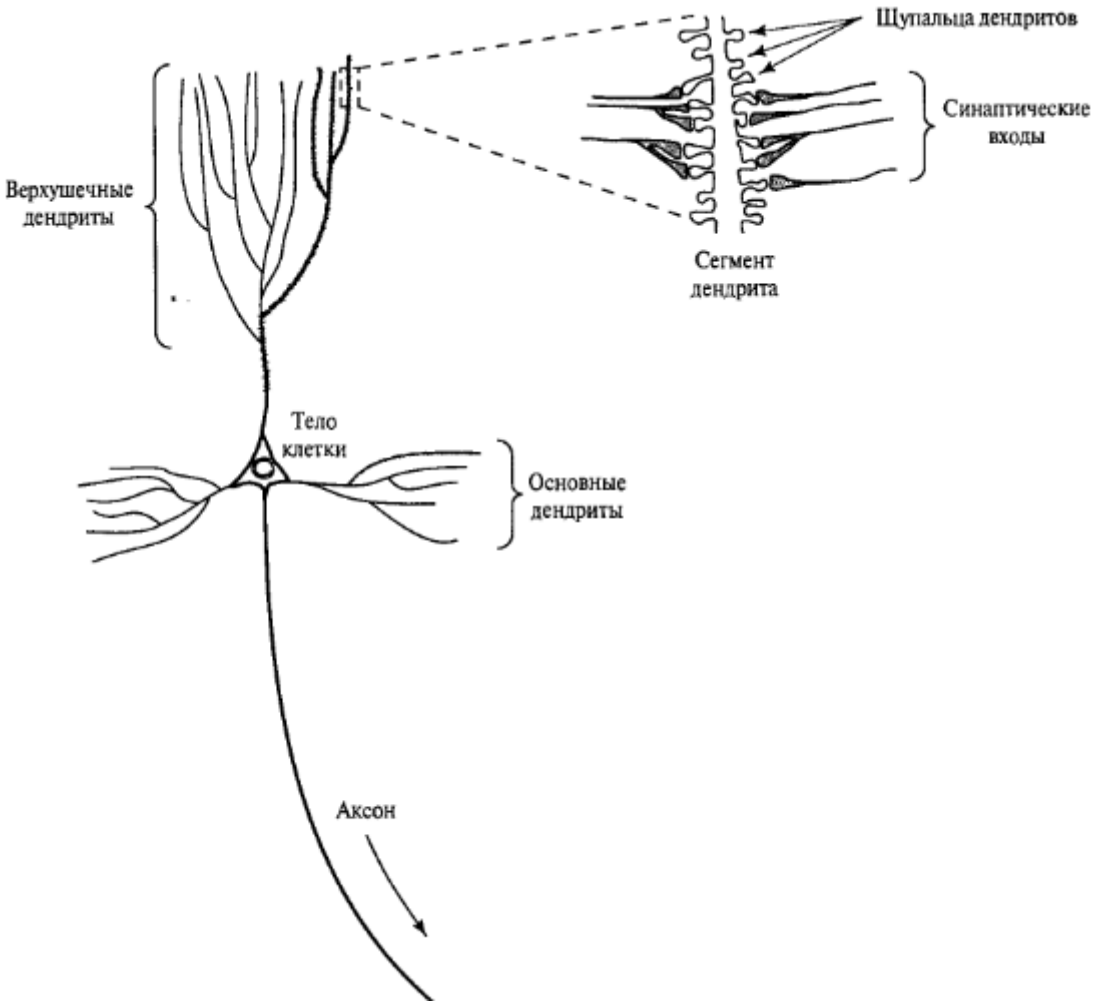

Рисунок 1 – Биологическое представление нейрона

Для нас же интерес представляет математическая модель нейрона [2] (рисунок 2), включает в себя следующее:

1) набор синапсов(связей), каждый синапс имеет собственный вес;

2) сумматор, который складывает входные сигналы;

функция активации, которая нормализует выходной сигнал.

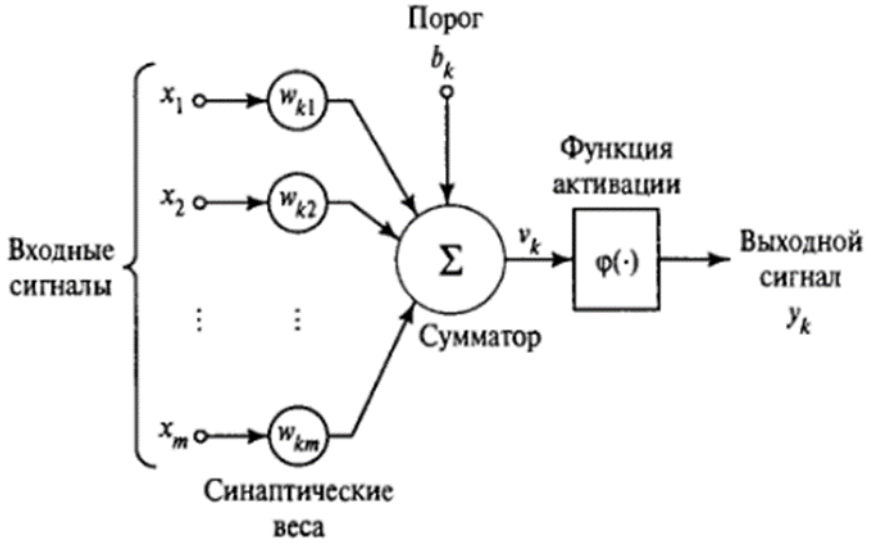

Рисунок 2 – Математическая модель нейрона

Существуют различные алгоритмы, которые помогают в написании текста самым известным является T9. Данный алгоритм позволяет предсказывать слова, которые следуют друг за другом, а также дописывать слова, написанные частично. Однако, у данного алгоритма имеется недостаток. Нет возможности добавлять свои слова в словарь.

При написании различной документации часто используются характерные сокращения, которых нет в исходном словаре. Но эту проблему можно решить, используя алгоритмы с возможностью самообучения. Такие алгоритмы позволяют адаптироваться к написанию различного рода документациям, а также добавлять новые слова в словарь.

В данной работе рассмотрим алгоритм, который позволяет добавлять новые слова в словарь.

Для демонстрации работы алгоритма разработана специальная программа. В качестве среды программирования выбрана Visual Studio и язык C# [4]. Поскольку в дальнейшем данный алгоритм планируется использовать в кроссплатформенном приложении, разработанном с использованием Unity. В этом случае Visual Studio предоставляет наиболее удобный инструментарий для разработки. Также поскольку для языка C#, отсутствуют завершенные рабочие библиотеки для работы с нейронными сетями, а у имеющихся тестовых библиотек функционал слишком избыточен.

Поэтому была разработана специальная библиотека для построения искусственных нейронных сетей с простой топологией.

Библиотека включает в себя два класса для работы с искусственной нейронной сетью:

1) класс Dendrite – позволяет соединять между собой нейроны;
2) класс Neuron – включает себя функционал для работы нейрона и его обучения.

Для обучения использовался алгоритм обратного распространения ошибки. Данный алгоритм является разновидностью алгоритмов, которые базируются на принципе обучения с учителем.

Алгоритм базируется на искусственной нейронной сети с топологией [3] (рисунок 3), которая позволяет проверять на совпадения слова. Сам алгоритм состоит из использования большого количества искусственных нейронных сетей для распознавания слов на вход подается вектор из текущих значений того на сколько текущий символ близок к тому, что

содержится в слове на данной позиции. Для создания такого вектора существует большое количество алгоритмов.

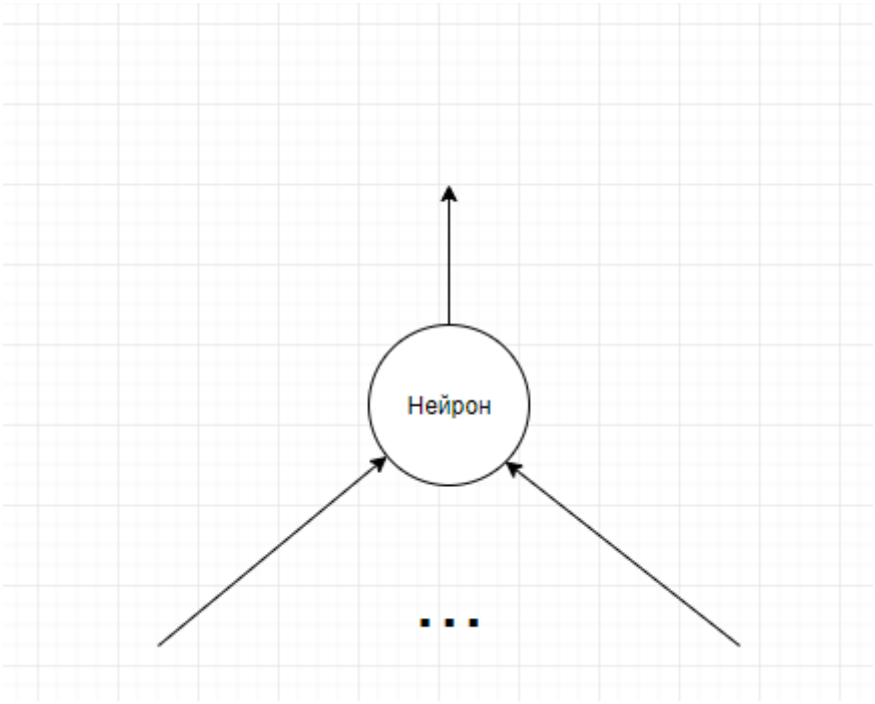

Рисунок 3 – Однослойная искусственная нейронная сеть прямого распространения

В данной работе мы используем простой алгоритм [1], которого вполне достаточно для тестового варианта.

$$\frac{C_i^p}{\left|C_i^p + \left|C_i^p - C_i^{in}\right|\right|} \quad (1)$$

Где C_i^p – числовое представление символа в слове. C_i^{in} – числовое представление входного символа.

Проверка работы алгоритма проведена на следующих тестах (рисунок 4).

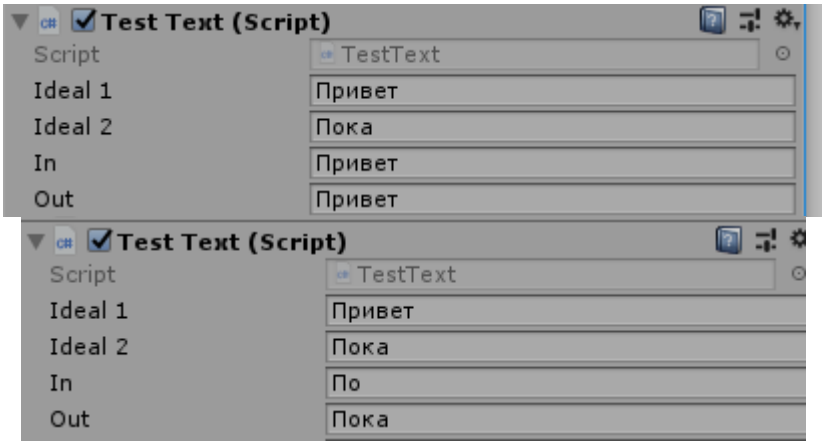

Рисунок 4 – Тесты

В результате тестирования были выявлены следующие ограничения алгоритма. Поскольку все символы имеют одинаковый вес в независимости от их позиций, а размер входного вектора зависит от размера слова более короткие слова имеют больший приоритет на срабатывания.

Вывод: данный алгоритм нельзя в полной мере использовать для добавления слов в словарь. Однако, в результате было выявлено несколько путей развития алгоритма, для того, чтобы начать его полноценно использовать. Самый очевидный путь добавление нормализации, чтобы слова с меньшим количеством букв имели одинаковый приоритет срабатывания с более длинными словами.

Список литературы

1. Хайкин Саймон Нейронные сети: полный курс, 2-е издание, ; Пер. с англ. – М. : Издательский дом “Вильямс”, 2006. – 1104 с. : ил. – Парал. тит. англ.
2. Фрэнк Розенблатт: Принципы нейродинамики. Перцептроны и теория механизмов мозга.
3. Рашид Т. Создаем нейронную сеть. : Пер. с англ. Санкт-Петербург. : ООО ”Альфа-книга”, 2017 – 272 с.
4. Шилдт Герберт C# 4.0: полное руководство.: Пер. с англ. – М.: ООО “И.Д. Вильямс”, 2011. – 1056 с.: ил. – Парал. тит. англ.
5. Мартин Р., Мартин М. Принцыпы, паттерны и методики гибкой разработки на языке C#. – Пер. с англ. – СПБ.: Символ-Плюс, 2011. – 768 с., ил.

References

1. Simon Haykin Neural networks: a complete course, 2nd edition ; TRANS. from English. – Moscow : Publishing house " Williams”, 2006. – 1104 p.: Il. – Paral. Titus. English.Volkova V. N, Denisov A.A.Theor of systems: - M: the Higher school, 2006. (in Russian)
2. Frank Rosenblatt: Principles of neurodynamics. Perceptrons and theory of brain mechanisms.
3. Rashid T. Creating a neural network. : Per. with English. Saint-Petersburg. : Alfa-kniga LLC, 2017 – 272 p.
4. Shield Herbert C# 4.0: a complete guide.: Per. with English. – M.: LLC “I. D. Williams”, 2011. – 1056c.: Il. – Paral. Titus. English.
5. Martin R., Martin M. Principles, patterns and techniques of agile development in language C#. – Per. with English. – SPb.: Plus Symbol, 2011. – 768 p., Il.

Бабак Н.Г., Крюков А.Ф. Защита информации в операционной системе Android // Международный журнал информационных технологий и энергоэффективности. – 2019. – Т.4 №1(11) с. 21-26

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.457

ЗАЩИТА ИНФОРМАЦИИ В ОПЕРАЦИОННОЙ СИСТЕМЕ ANDROID

[1]Бабак Н.Г., [2]Крюков А.Ф.

Федеральное государственное бюджетное образовательное учреждение высшего образования «Национальный исследовательский университет «МЭИ», Россия (111250, г.Москва, ул. Красноказарменная, д. 14); e-mail: [1]nikita.enrollee@gmail.com, [2]KriukovAF@mpei.ru

Описывается реализация аппаратных методов защиты данных в операционной системе Android. Сравнивается полнодисковое и файловое шифрование, приводятся преимущества и недостатки каждого метода. Описываются способы аутентификации пользователей в Android.

Ключевые слова: защита информации, Android, криптография, шифрование.

INFORMATION SECURITY IN THE ANDROID OPERATION SYSTEM

[1]Babak N.G., [2]Kryukov A.F.

National Research University “Moscow Power Engineering Institute”, Russia (111250, Moscow, Krasnokazarmennaya street, 14); e-mail: [1]nikita.enrollee@gmail.com, [2]KriukovAF@mpei.ru

The paper describes the implementation of hardware data protection methods in the Android operating system. Compares full-disk and file-based encryption, gives advantages and disadvantages of each method. Describes how to authenticate users in the Android.

Keywords: data protection, Android, cryptography, encryption.

Для людей всегда актуальна проблема защиты их личных данных. Поскольку мобильные устройства имеют большую популярность, то в предлагаемой статье рассмотрена реализация защиты данных в операционной системе Android.

Android – операционная система с открытым исходным кодом, основана на ядре Linux и собственной реализации виртуальной машины Java от Google. До версии Android 5.0 использовалась виртуальная машина Dalvik, а после – ART (Android Runtime) [1].

Существуют различные способы защиты информации на Android устройстве. К ним относятся, например, установка пароля, использование антивирусных программ, шифрование всего устройства, использование шифрующих приложений для хранения определённых данных. Наличие пароля в том или ином виде подразумевается во всех методах защиты информации.

Шифрование – это процесс кодирования данных пользователя на Android устройстве с помощью симметричных шифров. При записи данные сначала автоматически шифруются, а при чтении расшифровываются. Шифрование гарантирует, что злоумышленник не сможет прочесть данные даже при получении доступа к ним. [2]

В Android существует два метода шифрования:

- полнодисковое шифрование (full-disk encryption);
- файловое шифрование (file-based encryption).

Полнодисковое шифрование (FDE) Android устройства основано на модуле dm-crypt, входящем в функционал ядра Linux и обеспечивающем возможность шифрования на любом блочном устройстве хранения данных.

Полноценное полнодисковое шифрование стало возможным с выходом версии 5.0, что связано с появлением 64-битных процессоров.

При первом включении устройство генерирует случайный 128-битный мастер-ключ (device encryption key или DEK), а затем хэширует его паролем по умолчанию и солью. Пароль задаётся пользователем, а соль – сгенерированное устройством 128-битное случайное число.

С помощью DEK шифруются все данные. Пользователь не видит и не использует этот ключ. Когда пользователь меняет пароль, то заново шифруется только хранимый DEK ключ, а не все данные.

Если при первоначальном шифровании устройство потеряет питание, то имеющиеся данные будут утеряны и потребуется сброс к заводским настройкам.

Алгоритм шифрования ключа DEK перед его сохранением в криптографические метаданные представлен на рисунке 1 [3].

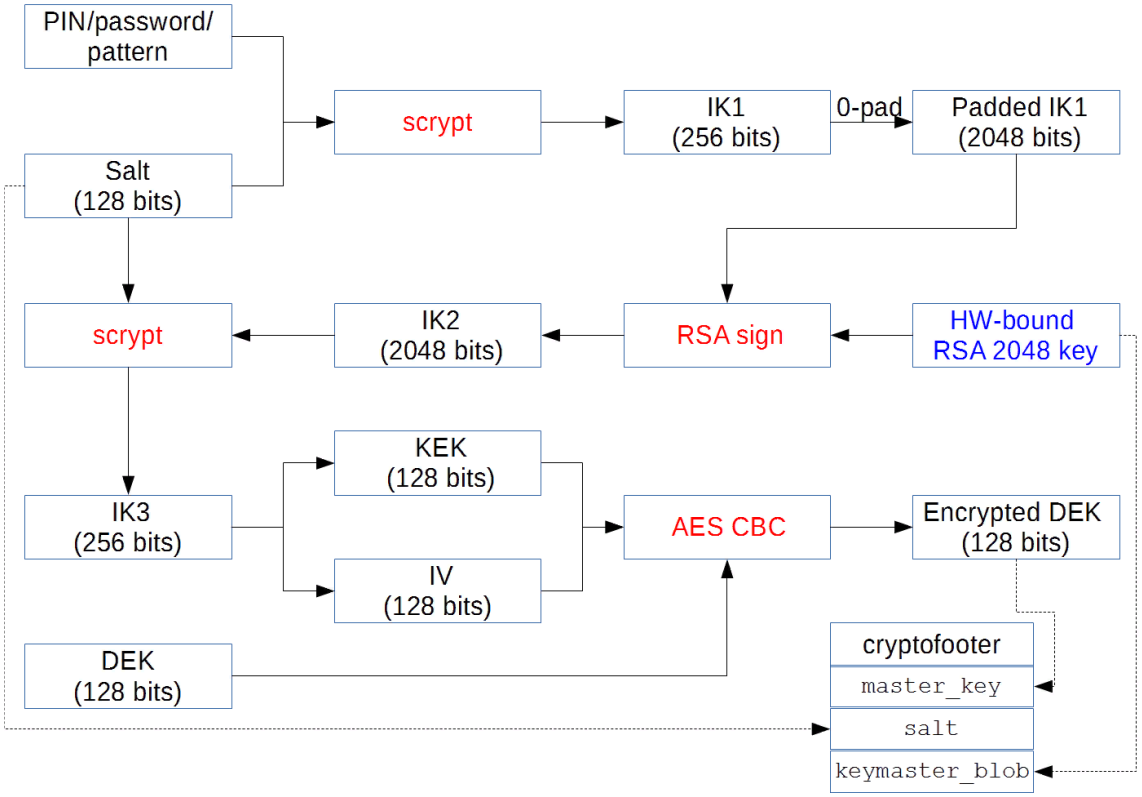

Рисунок 1 – Алгоритм шифрования ключа DEK

AES (Advanced Encryption Standard) – симметричный алгоритм блочного шифрования, принятый правительством США на основе результатов проведенного конкурса в качестве стандарта шифрования. Основу алгоритма составляют замены, подстановки и линейные преобразования, каждое из которых выполняется блоками по 128 бит.

CBC (Cipher block chaining) – режим сцепления блоков шифротекста – один из режимов шифрования для симметричного блочного шифра с использованием механизма обратной связи. Каждый блок открытого текста (кроме первого) побитно складывается по модулю два с предыдущим результатом. Одна ошибка в бите блока шифротекста влияет на расшифровку всех последующих блоков. Перестройка порядка блоков зашифрованного текста вызывает повреждения результата дешифрования [4]. На рисунке 2 представлена схема работы режима CBC.

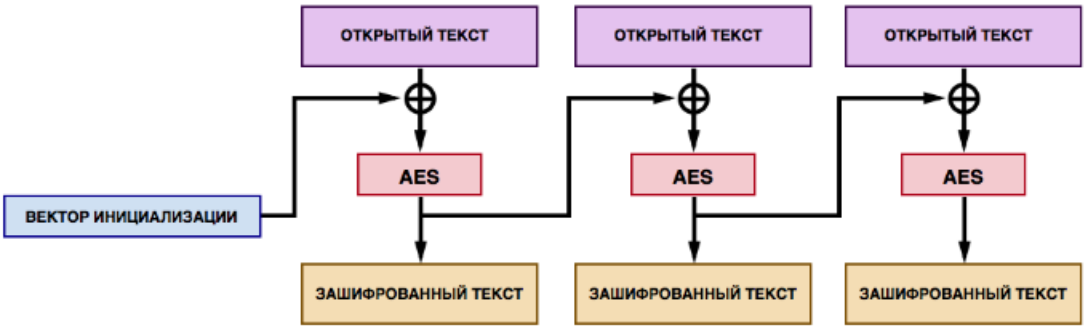

Рисунок 2 – Режим CBC

Важно отметить, что криптографическая схема AES-CBC считается уязвимой к утечке данных, так как допускает определение точки их изменения. Она позволяет выполнять атаки по типу подмены и перемещения.

На данный момент Android уходит от поддержки полнодискового шифрования и советует использовать файловое шифрование. Одна из причин такого решения – невозможность осуществления экстренного вызова после перезагрузки устройства, пока пользователь не введёт пароль.

Файловое шифрование (FBE) стало доступно, начиная с версии Android 7.0. Файловое шифрование, которое выполняется с использованием возможностей файловой системы ext4, позволяет шифровать различные файлы различными ключами и расшифровывать их независимо.

Файловое шифрование добавляет новую функцию Direct Boot [5] (прямая загрузка). Прямая загрузка позволяет работать приложениям, когда устройство было включено, но не разблокировано пользователем. Режим Direct Boot по умолчанию не включен в приложении и при включении удаляет все имеющиеся данные. Ранее в Full-disk encryption было необходимо ввести пароль, прежде чем получить доступ к каким-либо функциям устройства.

С введением файлового шифрования приложения могут работать в зашифрованном режиме, что позволяет защитить личные данные пользователя именно там, где это действительно необходимо.

На устройствах с File-based encryption имеется два доступных для приложений хранилища:

- шифрование на уровне учётных данных – Credential Encrypted (CE);

- шифрование на уровне устройства – Device Encrypted (DE).

CE хранилище используется по умолчанию и доступно только после разблокировки устройства. DE хранилище доступно в режиме Direct Boot и после того, как пользователь разблокировал устройство.

При отключенном файловом шифровании хранилища DE и CE всегда находятся в разблокированном состоянии. Direct Boot позволяет приложениям обращаться к каждому из этих хранилищ.

Такое разделение хранилищ делает рабочие профили более безопасными, поскольку шифрование больше не основано исключительно на пароле загрузки, как в Full-disk encryption.

Содержимое файлов шифруется с помощью шифра AES-256 в режиме XTS. Имена файлов шифруются шифром AES-256 в режиме CBC-CTS. Режим XTS разрабатывался специально для шифрования на блочных устройствах и не имеет типичных для режима CBC уязвимостей. В частности, XTS не позволяет определить точку изменения данных, не подвержен утечке данных, устойчив к атакам подмены и перемещения.

Android поддерживает следующие способы аутентификации пользователя:

- ПИН-код;
- пароль;
- графический ключ;
- отпечаток пальца.

При первом запуске устройства пользователь вводит ПИН-код, пароль или графический ключ. Эта начальная регистрация создаёт случайно сгенерированный 64-разрядный идентификатор безопасности пользователя (SID – user secure identifier). SID привязан к паролю.

После настройки пользователем учётных данных, он получает идентификатор SID и может приступить к аутентификации, которая начинается с ввода ПИН-кода, пароля, графического ключа или с помощью отпечатка пальца.

Все компоненты безопасной среды исполнения (trusted execution environment - TEE) имеют общий секретный ключ, используемый для аутентификации.

На рисунке 3 представлена схема процесса проверки подлинности [6].

Процесс аутентификации.

1. Пользователь вводит ПИН, пароль, графический ключ или отпечаток пальца. И в зависимости от метода проверки отправляется запрос в gatekeeperd или fingerprintd, называемые деймон (daemon).
2. Деймон посылает данные своей дочерней части в TEE, которая генерирует токен аутентификации AuthToken.
3. Деймон получает подписанный AuthToken и передаёт его службе хранилища ключей.
4. Служба хранилища ключей передаёт AuthToken мастеру ключей keymaster и проверяет подлинность ключа.

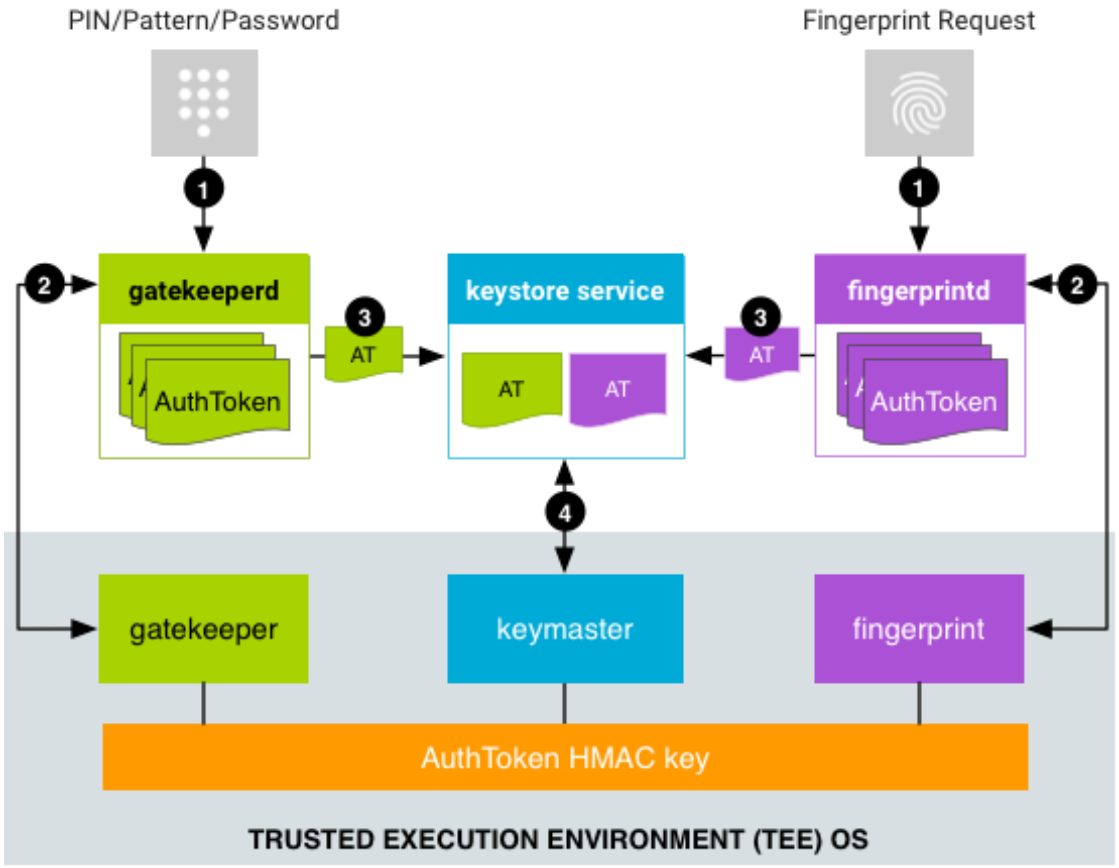

Рисунок 3 – Процесс проверки подлинности

AuthToken после перезагрузки устройства становится недействительным. Формат AuthToken'а унифицирован и состоит из следующих компонентов:

- версия токена аутентификации – 1 байт;
- идентификатор операции – 64 бита;
- неповторяющийся идентификатор пользователя, привязанный ко всем ключам – 64 бита;
- идентификатор проверки подлинности ASID – 64 бита;
- тип аутентификатора (Gatekeeper или Fingerprint) – 32 бита;
- временная метка, время в миллисекундах с момента последней загрузки системы – 64 бита;
- AuthToken HMAC – 256 бит.

При каждой загрузке устройства случайным образом генерируется ключ AuthToken HMAC, который сообщается всем компонентам безопасной среды исполнения TEE (Gatekeeper, Fingerprint). Данный ключ не должен быть доступен за пределами безопасной среды исполнения [6].

Файловое шифрование не лишено недостатков. Данный метод уязвим к side channel [7] атакам, так как, несмотря на шифрование файлов и их имен, он оставляет открытыми метаданные, которые можно использовать для выяснения типа хранимой информации и идентификации пользователя устройства. Также встроенные методы шифрования приводят к снижению производительности. Но лучше использовать их, чем хранить данные в сторонних непроверенных приложениях.

Учитывая преимущества и недостатки каждого метода, сделан вывод, что при поддержке устройством файлового шифрования следует использовать его. Полнодисковое шифрование имеет смысл только на устройствах с операционной системой Android версии меньше 7.0. При необходимости защиты определённых данных можно использовать проверенные, желательно сертифицированные, приложения, созданные специально для защиты данных. Это позволит уменьшить нагрузку на устройство, а значит увеличить его производительность, также это даёт возможность учитывать специфику шифрования тех или иных типов данных.

Список литературы

1. Android. URL: https://ru.wikipedia.org/wiki/Android
2. Encryption. URL: https://source.android.com/security/encryption
3. Извлечение аппаратного ключа полнодисковой защиты в телефонах Android на процессорах Qualcomm. URL: https://sohabr.net/habr/post/395643
4. AES шифрование и Android клиент. URL: https://habr.com/company/rambler-co/blog/279835
5. Режим Direct Boot. URL: https://developer.android.com/training/articles/direct-boot
6. Authentication. URL: https://source.android.com/security/authentication
7. Атака по сторонним каналам. URL: https://ru.wikipedia.org/wiki/Атака_по_сторонним_каналам

References

1. Android. URL: https://ru.wikipedia.org/wiki/Android
2. Encryption. URL: https://source.android.com/security/encryption
3. Извлечение аппаратного ключа полнодисковой защиты в телефонах Android на процессорах Qualcomm. URL: https://sohabr.net/habr/post/395643
4. AES шифрование и Android клиент. URL: https://habr.com/company/rambler-co/blog/279835
5. Режим Direct Boot. URL: https://developer.android.com/training/articles/direct-boot
6. Authentication. URL: https://source.android.com/security/authentication
7. Атака по сторонним каналам. URL: https://ru.wikipedia.org/wiki/Атака_по_сторонним_каналам