

*Б.Р.Авхадеев
Л.И.Воронова
Е.П.Охалкина
Москва, Россия*

*B.R.Avkhadeev
L.I.Voronova
E.P.Okhapkina
Moscow, Russia*

РАЗРАБОТКА РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ ДАННЫХ ИЗ ПРОФИЛЯ СОЦИАЛЬНОЙ СЕТИ «ВКонтакте»

DEVELOPING A RECOMMENDER SYSTEM BASED ON THE SOCIAL NETWORK «VKontakte» PROFILE DATA

Аннотация. В статье рассматривается проблема автоматизации процесса веб-сёрфинга и фильтрации контента. Автоматизация процесса веб-сёрфинга с фильтрацией отображаемого контента на основе онтологий конкретных предметных областей — весьма интересная и нетривиальная задача, которая решается в рамках направления, связанного с разработкой рекомендательных систем (РС).

В статье описано проектирование и реализация мультиагентной рекомендательной системы «EZSurf», обеспечивающей анализ интересов и предоставление рекомендаций пользователям социальной сети «ВКонтакте» на основе данных из профиля конкретного пользователя.

Проанализированы методы, технологические решения, продукты-аналоги, связанные с проблемой фильтрации контента, и сформулированы требования к системе. Рекомендательная система «EZSurf» автоматизирует процесс веб-сёрфинга и фильтрации контента, используя профиль пользователя в конкретной социальной сети для сбора данных и API сторонних сервисов (LastFM, TheMovieDB) для получения сведений о схожих объектах. Такой подход существенно оптимизирует систему, поскольку не требует создания собственной системы классификаций и базы объектов.

В рамках мультиагентной системы реализовано три агента. Первый агент (сборщик) собирает данные из профиля пользователя «ВКонтакте» посредством API. Второй агент (анализатор), также посредством API, собирает данные о схожих объектах из сторонних сервисов. Для поиска контента реализован третий агент по принципу поискового робота (рекомендатель).

Система «EZSurf» может применяться в повседневной жизни пользователями социальной сети «ВКонтакте» для уменьшения времени, затрачиваемого на поиск в Интернете интересующей их информации, при этом получая рекомендации, отфильтрованные в зависимости от предпочтений каждого конкретного пользователя.

Система имеет возможности дальнейшего развития, среди которых можно выделить увеличение реестра сайтов; развитие онтологий предметных областей, оптимизацию алгоритма поискового агента-рекомендателя для индексации страниц и парсинга их содержимого, сбор данных из профиля в нескольких социальных сетях.

Ключевые слова: мультиагентная; рекомендательная система; система; фильтрация содержимого; социальные сети; рекомендательный контент; веб-сёрфинг.

Abstract. The following article considers the problem of web-surfing automation and content filtration. The principal objective of this project is to develop a software solution to this problem - a multi-agent system for analyzing VKontakte users' interests - and providing a recommendation system EZSurf.

The article describes the development and application of a multi-agent recommender system EZSurf that performs analysis of interests and provides recommendations for the social network VKontakte users based on the data from the profile of a particular user.

The article also provides an analysis of different methods, technological solutions, and similar products aimed at content filtration, as well as their advantages and disadvantages. EZSurf allows automating the web-surfing process and content filtration with the use of user's profile in a particular social network to collect data and API of external services (LastFM, TheMovieDB). For search and selection of information an agent (Recommender) that works as web-crawler has been implemented. Such an approach contributes to optimization of the recommender system, because it does not require creation of its own object classification system and objects database.

The functionality of multi-agent system was separated between three agents. The first agent (Collector) collects user data from «VKontakte» profile using VK API. The second agent (Analyzer) collects similar objects from databases of external services (LastFM, TheMovieDB) that will be the criteria for further search of recommendatory content. The third agent (Recommender) based on the principle of a search robot is used for searching content.

System «EZSurf» can be exploited by the users of social network «VKontakte» in everyday life to save time on web-surfing. At the same time the users will get recommendations on the content filtered depending on preferences of every particular user.

The system can be further developed. There are several ways for its evolution: extension of sites registry, usage of other data from profile in addition, optimization of the algorithm for web-pages indexing and content parsing (Recommender), data collection from the several social networks.

Key words: multi-agent system; system; recommender system; content filtration; social networks; recommendatory content; web-surfing.

Сведения об авторах: Авхадеев Булат Ринатович¹, студент факультета программной инженерии; Воронова Лилия Ивановна², заведующая кафедрой информационных систем и моделирования; Охажкина Елена Павловна³, старший преподаватель кафедры информационных систем и моделирования.

Место работы: ¹ Национальный исследовательский университет «Высшая школа экономики»; ^{2,3} Российский государственный гуманитарный университет.

About the author: Bulat Rinatovich Avkhadeev¹, student of the Department of Software Engineering; Lilia Ivanovna Voronova², Head of the Department of Information Systems and Modeling; Elena Pavlovna Okharkina³, Senior Lecturer, Department of Information Systems and Modeling.

Place of employment: ¹ National Research University «Higher School of Economics»; ^{2,3} Russian State University for the Humanities.

Контактная информация: 111672, г. Москва, ул. Новокосинская, д. 40, кв. 204; тел.: 9160686838.
E-mail: voronova2001@mail.ru

Многие пользователи Интернета встречались с такой проблемой, как поиск интересной, актуальной информации. Зачастую этот процесс, схематично приведенный на рис. 1, отнимает немало времени для того, чтобы найти именно то, что нужно.

Автоматизация процесса веб-сёрфинга с фильтрацией контента от неподходящей информации — весьма интересная и нетривиальная задача, которая решается с помощью рекомендательных систем (РС).



Рис. 1. Процесс веб-сёрфинга

Рекомендательные системы (РС) — новое направление информационных технологий, связанное с выявлением предпочтений пользователей.

РС — программы, которые пытаются предсказать, какие объекты (фильмы, музыка, книги, новости, товары, веб-сайты) будут интересны пользователю, имея определенную информацию о его профиле.

Существует несколько основных алгоритмов РС:

- *Коллаборативная фильтрация* — метод построения рекомендательных систем, основанный на предположении о том, что пользователям со схожими оценками к предметам, просмотренным ранее, будет нравиться одно и то же в будущем. Группа пользователей, наиболее схожая по интересам с активным, называется «соседями» [5; 7].

- *Фильтрация содержимого* — при данном подходе для каждого пользователя создается профиль в системе, который хранит историю его оценок объектам этой системы. Система выдает пользователю предметы, схожие с теми, которые ему понравились в прошлом [7].

- *Гибридные рекомендательные системы* — строятся с использованием комбинации подходов, перечисленных выше. Такие системы направлены на то, чтобы компенсировать недостатки одного подхода внедрением другого [7].

Каждый из подходов обладает некоторыми недостатками, которые могут помешать при выполнении поставленной задачи. К примеру, рекомендательные системы, использующие метод коллаборативной фильтрации, зачастую сталкиваются с проблемой «холодного старта», ситуацией, когда в системе появляется новый пользователь и нет никакой информации о том, что ему нравится [8]. В частности, в социальной сети «ВКонтакте» ее политика ограничивает доступ к данным друзей пользователя, возникает сложность определения «соседей», что делает использование подхода коллаборативной фильтрации невозможным.

Для определения функциональности системы «EZSurf» проведен сравнительный анализ проектов-аналогов, схожих по функциональности с текущим проектом, и различных сервисов, использующих рекомендательные системы.

Главным аналогом является российский сервис Surfingbird с интегрированной рекомендательной системой, который занимается подборкой персонально интересных веб-страниц [9]. Таким образом, Surfingbird тоже подбирает контент на основе рекомендаций и автоматизирует процесс веб-сёрфинга, однако разница в том, что данный сервис использует для сбора данных опрос нового пользователя, чтобы избежать проблемы «холодного старта», и в дальнейшем опирается на оценки пользователя.

В социальной сети «ВКонтакте» есть встроенные рекомендации (рис. 2), но распространяются они только на музыку, и далеко не всем нравится, как этот сервис работает.

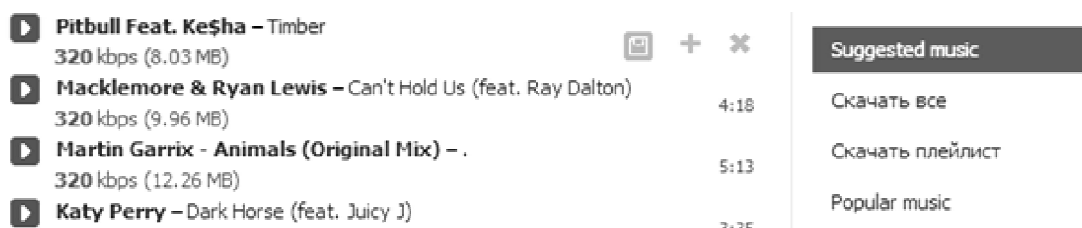


Рис. 2. Пример рекомендации музыкального контента «ВКонтакте»

Ниже приведены примеры интеграции рекомендательных систем в различные проекты:

- *IMDb.com*

Сайт IMDb.com — крупнейшая в мире база данных кинофильмов, сериалов и телешоу. Под каждым фильмом можно найти раздел «People who liked this also liked...», что значит, что эти фильмы рекомендуют вам на основании схожести ваших вкусов со вкусом других людей. Очевидно, что в данном сервисе используется коллаборативная фильтрация, основанная на принципе поиска «соседей».

- *Amazon.com*

Крупнейший интернет-сервис продажи товаров. Пройдя регистрацию и начав просматривать товары, пользователь заполняет историю активности своего профиля, затем под каждым товаром можно найти рекомендации по тому же принципу, что и у IMDb: «Customers Who Bought Items in Your Recent History Also Bought».

- *Grooveshark*

Музыкальный веб-сайт с потоковым мультимедиа и интегрированной рекомендательной системой, позволяющий пользователям искать, слушать, загружать и сохранять музыку бесплатно. Одной из особенностей сайта является его рекомендательный сервис, подбирающий песни, похожие на те, которые находятся в очереди воспроизведения у пользователя. Специальной иконкой (смайликом) можно сообщить сервису, была ли та или иная рекомендация удачной. В данном сервисе используется рекомендательная система, основанная на фильтрации содержимого [6].

По результатам анализа предметной области [3; 4] были сформулированы функциональные требования к РС «EZSurf»:

1. Система строится на основе мультиагентной архитектуры.
2. Авторизация в системе «EZSurf» происходит через социальную сеть «ВКонтакте».
3. РС не производит каких-либо действий до прохождения корректной авторизации.
4. При первом запуске РС «EZSurf» после успешной авторизации система должна потребовать разрешения пользователя на обработку данных с его страницы «ВКонтакте».
5. Система выполняет сбор аудио-, видеоданных, а также данных об интересах из профиля пользователя «ВКонтакте».
6. Система хранит собранные данные в базе данных.

7. Система производит ранжирование аудиоданных по количеству их встречаемости на странице «ВКонтакте».

8. Система находит аудио- и видеообъекты (рекомендации), схожие с теми, что были собраны из профиля «ВКонтакте», посредством обращения к сторонним базам и API (lastFM, IMDb).

9. Система производит индексацию веб-страниц для каждого сайта из реестра, созданного модератором.

10. Система производит поиск рекомендаций на проиндексированных страницах.

11. Система должна отображать конечный результат работы в виде рекомендации, содержащей заголовок, изображение и краткое описание статьи.

В мультиагентной архитектуре РС «EZSurf» выполнение функциональных задач распределяется между разными агентами.

1) Задача сбора информации.

Сборщик — агент, собирающий информацию из профиля пользователя сети «ВКонтакте». Это такая информация как статус, оценки «мне нравится», сообщества, аудиозаписи, видео, всё, что может помочь выявить интересы пользователя.

2) Задача выявления того, что пользователю нравится больше из того, что есть у него на странице и определение рекомендаций по выявленным интересам.

Анализатор — агент, осуществляющий анализ контента со страницы пользователя, определение предпочтений и формирование рекомендаций по содержанию контента.

3) Задача поиска и сбора рекомендуемого контента в сети Интернет для последующего предоставления пользователю.

Рекомендатель — агент, собирающий подходящий контент в сети Интернет, основываясь на рекомендациях для каждого конкретного пользователя.

На рис. 3 приведена архитектура мультиагентной системы, обеспечивающая решение перечисленных задач и заявленную функциональность агентов.

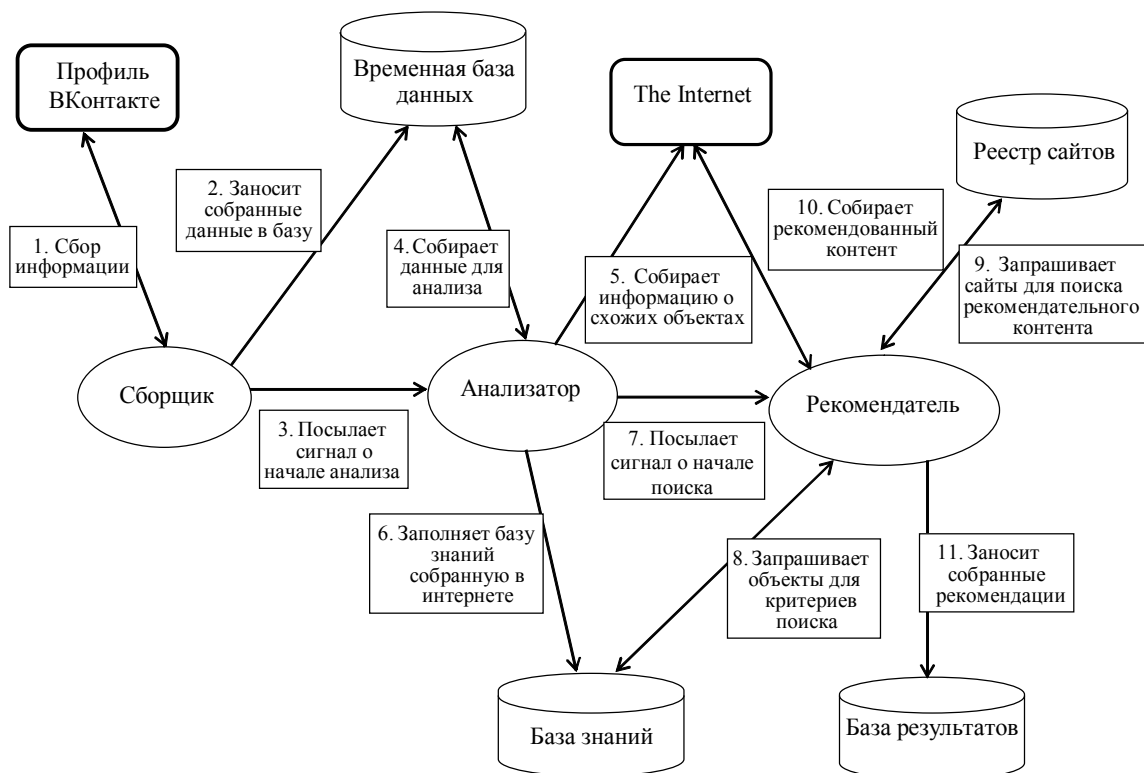


Рис. 3. Архитектура системы «EZSurf»

Профиль «ВКонтакте» подразумевает все данные со страницы пользователя. Страница пользователя «ВКонтакте» содержит информацию разного рода: персональная информация, которую пользователь заполняет сам, список сообществ, подписок, который формируется в ходе активности на сайте ВКонтакте, медиа-контент (фото, аудио, видео).

Агент *Сборщик*, посредством вызова методов ВКонтакте API, получает ответ в формате JSON, откуда он вычленяет необходимую информацию. Например, в случае с аудиозаписями из запроса ВКонтакте API `audio.get` (рис. 4), *Сборщик* извлечет только тег *artist*, так как именно по *артисту* будут выдаваться рекомендации. Затем *Сборщик* заносит собранные данные в базу данных для дальнейшего анализа.

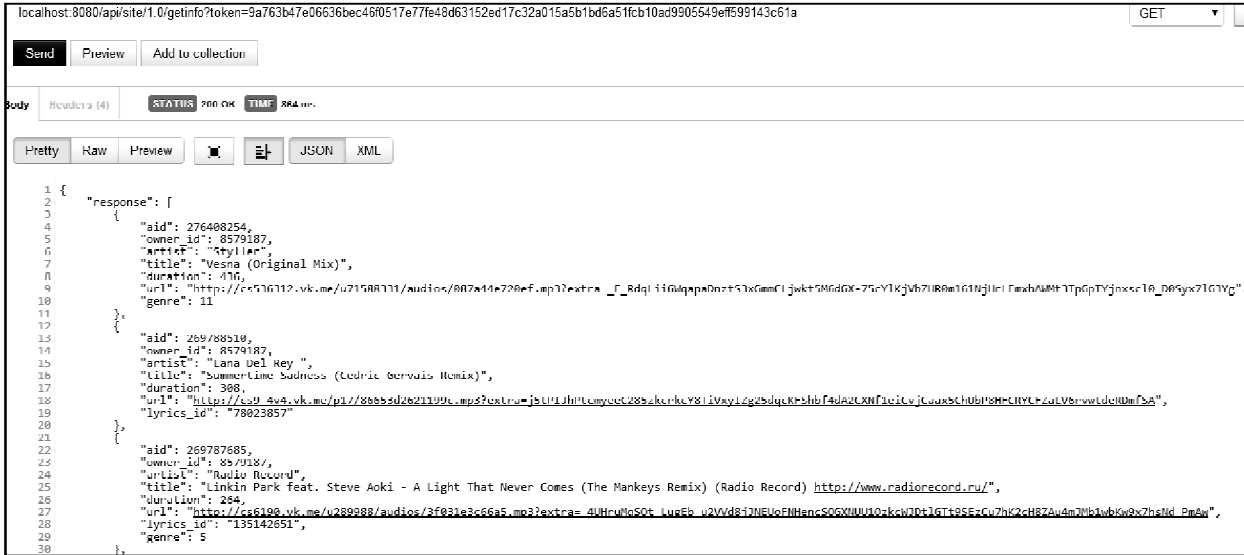


Рис. 4. Ответ в формате JSON на запрос `audio.get` VK API

Когда *Сборщик* закончил сбор данных, он подает сигнал *Анализатору*. Агент *Анализатор*, в первую очередь, производит ранжирование по данным, собранным *Сборщиком*. Это позволит в дальнейшем расставить приоритеты для рекомендаций (более или менее интересные). Для ранжировки объектов по важности используется вариация метода количественного контент-анализа. В случае контент-анализа по количеству встречаемых слов в тексте определяется тематика, например, статьи в газете. По такому же принципу можно оценить, какой исполнитель интересует пользователя больше других по количеству песен каждого исполнителя в списке аудиозаписей из профиля социальной сети «ВКонтакте». После ранжирования *Анализатор*, путем обращения к сторонним базам, имеющим классификацию объектов (IMDb, lastFM), выявляет объекты, схожие с теми, что были собраны со страницы пользователя «ВКонтакте», которые будут являться критериями поиска рекомендаций.

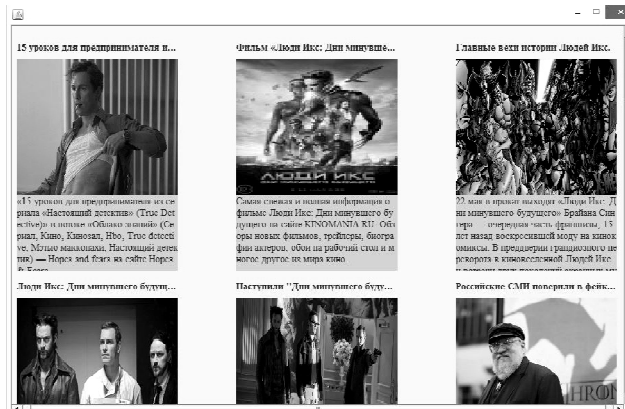


Рис. 5. Пример работы программы — PC «EZSurf»

Закончив определение предпочтений пользователя и выявив объекты для рекомендаций, *Анализатор* подает сигнал агенту *Рекомендатель*. *Рекомендатель* в свою очередь обращается к реестру сайтов, где он получает список сайтов, на которых может быть найден рекомендуемый контент. Реестр сайтов сократит масштаб поиска и индексирования множества лишних страниц. *Рекомендатель* индексирует каждый сайт из реестра и записывает список страниц каждого сайта в базу данных. Затем, имея список URL для поиска и объекты, которые могут заинтересовать активного пользователя, *Рекомендатель* собирает в Интернете контент и предоставляет его пользователю в виде, приведенном на рис. 5.

Для реализации системы «EZSurf» была использована среда разработки Netbeans IDE 8.0 и язык программирования Java. На рис. 6 представлена диаграмма классов системы «EZSurf». Для создания агентов использовалась сторонняя библиотека JADE (Java Agent Development Environment), каждый агент представлен отдельным классом, который наследует класс `jade.core.agent`.

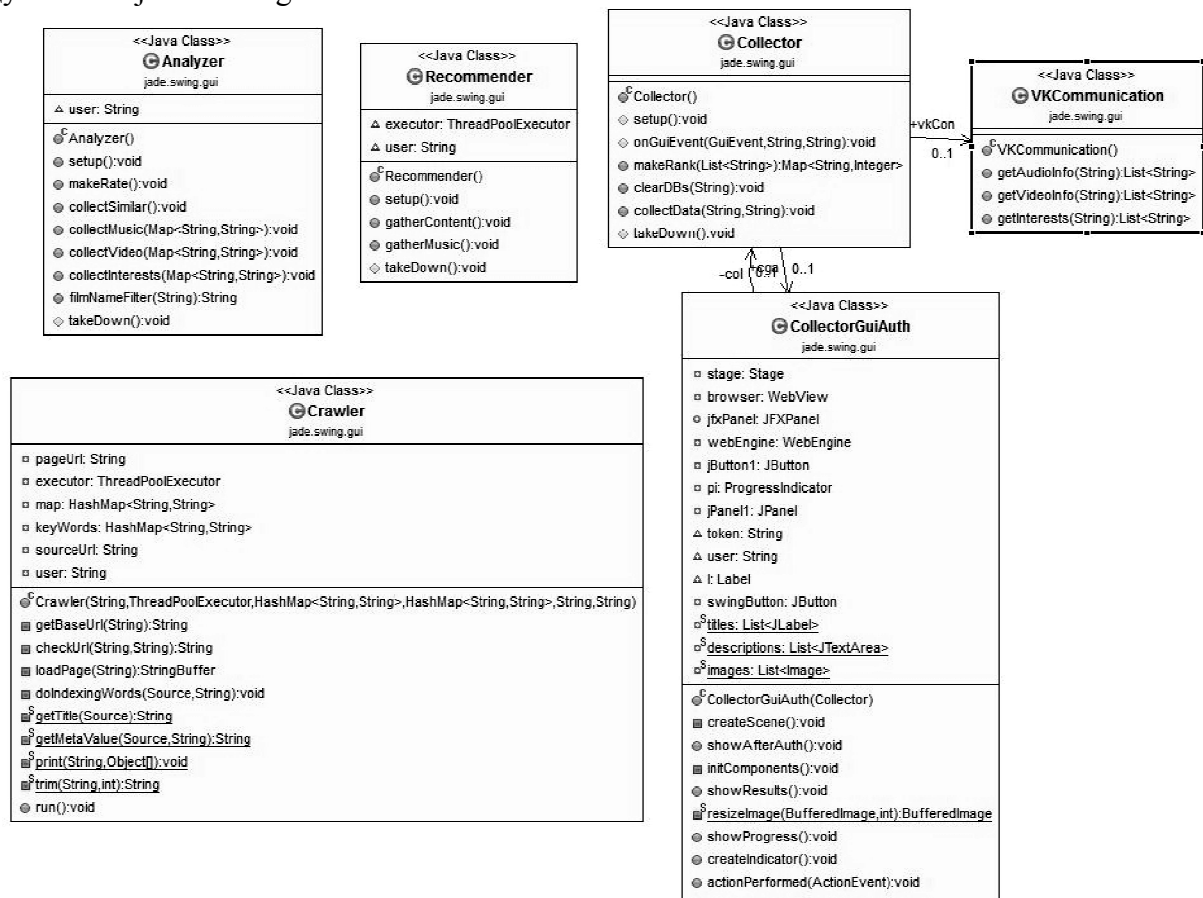


Рис. 6. Диаграмма классов системы «EZSurf»

Все агенты инициализируются при запуске программы. Происходит вызов метода `setup` всех трех агентов. Дальнейшее взаимодействие происходит посредством обмена ACL сообщениями, где JADE создает очередь и управляет потоком ACL-сообщений [1].

Агент *Сборщик*, представленный классом `Collector`, в котором происходит сбор данных из профиля пользователя в социальной сети «ВКонтакте», также отвечает за создание окна приложения в методе `setup`, который вызывается при создании агента. После того как пользователь авторизуется «ВКонтакте» и нажмет кнопку «SURF», сработает обработчик событий `actionPerformed` в классе `CollectorGuiAuth`, в котором происходит вызов метода агента `onGuiEvent`, который в свою очередь производит вызов метода сбора данных (`collectData`). Перед тем как начать собирать новые данные, метод `clearDBs` удаляет данные из базы по текущему пользователю.

Метод сбора данных использует объект класса VKCommunication, содержащий методы вызова VK API для получения списков аудио-, видео- и персональной информации, в которых также происходит парсинг JSON ответов для получения нужных данных. В случае с аудио это исполнитель.

По окончании сбора данных происходит отправление ACL сообщения агенту *Анализатору* при помощи средств библиотеки JADE, класса ACL Message.

Агент *Анализатор*, представленный классом Analyzer на диаграмме классов (рис. 6), производит вызов методов API LastFM и TheMovieDB для сбора схожих объектов для аудио и видео соответственно. Затем *Анализатор* посылает ACL сообщение-сигнал *Рекомендателю*. На рис. 7 виден результат работы *Анализатора*, во второй колонке содержится информация об объекте, в третьей колонке — id объекта, для которого данный объект является схожим, и в четвертой колонке — тип объекта.

136	7cb02ce8e4c	Black Sabbath	793481d6e4c611e391cfaba	audio	252880058
137	7cbc18a0e4c	Whitesnake	793481d6e4c611e391cfaba	audio	252880058
138	7cc5a1fee4c	Dio	793481d6e4c611e391cfaba	audio	252880058
139	7cd3f006e4c	Ian Gillan	793481d6e4c611e391cfaba	audio	252880058
140	7cdd7964e4c	Led Zeppelin	793481d6e4c611e391cfaba	audio	252880058
141	7ce96512e4c	Nazareth	793481d6e4c611e391cfaba	audio	252880058
142	7cf2ee70e4c	UFO	793481d6e4c611e391cfaba	audio	252880058
143	7cfeda28e4c	Glenn Hughes	793481d6e4c611e391cfaba	audio	252880058
144	7d0ac5d6e4c	deep purple	793481d6e4c611e391cfaba	audio	252880058

Рис. 7. Таблица recommendation

Получив сообщение о начале поиска, *Рекомендатель* (класс Recommender) — по сути, поисковый робот — производит индексацию каждого сайта из реестра, где реестр представляет собой таблицу базы данных, содержащую URL сайта и тип контента, который может быть найден на этом сайте (Рис. 8).

	site_id [PK] character	site_url character varying(255)	type character varying(50)
1	1d8de54edec	http://www.kinomania.ru/blog/	2
2	37e38e0cdec	http://instuff.ru/	3
3	4ef8cb56e3f	http://www.kinomania.ru/article/press_review/	2
4	51e06dbae3f	http://www.kinomania.ru/article/reason/	2
5	5c46d46ae3f	http://www.kinomania.ru/reviews/	2
6	70407ae8e3f	http://www.kinomania.ru/news/	2
7	76cade32deb	http://www.afisha.ru/movies/movie_list/cinemasoon/	2
8	8e7ffedee07	http://www.afisha.ru/msk/concerts/	1
9	9329222edec	http://www.lookatme.ru/	3
10	9ebaed8edec	http://www.hopesandfears.com/	3
11	df506efadec	http://tjournal.ru/	3
12	f79bfad6e07	http://www.afisha.ru/movies/movie_list/cinematoday/	2

Рис. 8. Таблица registry

В таблице registry используется следующая зависимость (тип контента), позволяющая расшифровать значения поля type: 0 (Интересы), 1 (Музыка), 2 (Видео), 3 (Все типы).

Индексация страниц происходит в классе Crawler по следующему алгоритму: загрузить страницу с адресом pageURL; поместить адрес pageURL в список посещенных страниц; найти ссылки на загруженной странице с адресом pageURL; для каждой найденной ссылки, не находящейся в списке посещенных страниц, запустить процедуру из пункта 1 [7].

Алгоритм использует рекурсивный вызов, что означает большую трудоемкость и ресурсоемкость, которые можно понизить параллельным запуском алгоритма в разных потоках. Для управления заданиями и потоками использованы такие структуры данных как пул потоков — `ThreadPoolExecutor` и связанная блокирующая очередь — `LinkedBlockingQueue` [2] из стандартной библиотеки `java.util`, что также позволяет оптимизировать данный процесс. Если создано заданий больше, чем может пул потоков обработать одновременно, то задачи простаивают в этой очереди, и время их жизни задается в конструкторе пула потоков. В данном случае время жизни заданий составляет 5 минут. За хранение адресов уже посещенных страниц отвечает такая структура данных как `hash map` из той же библиотеки `java.util`.

В процессе каждая страница проверяется на наличие рекомендательного контента. Из таблицы `recommendation` в структуру данных `List<String>` записывается колонка `rec_info`. В `rec_info` будет записано имя артиста, название видео или хобби, в случае, если в колонке `type` значение `audio`, `video` или `hobby` соответственно. Для каждого объекта в этом `List<String>`, выявленного *Анализатором* как рекомендуемого, происходит поиск совпадений на странице. С помощью сторонних библиотек `Jericho` и `JSoup` происходит извлечение содержимого метатегов `title`, `description` и `keywords`. Если хотя бы в одном из метатегов присутствует вхождение искомой строки и хотя бы одно вхождение найдено в теле HTML страницы, т.е. внутри тега `body`, то страница принимается за рекомендуемую. Содержимое тегов `title` и `description` записывается в базу результатов как заголовок и описание статьи соответственно.

Таким образом, спроектирована и реализована минимально нагруженная рекомендательная система, реализующая метод фильтрации содержимого (*content-based*), использующая профиль пользователя социальной сети «ВКонтакте». Это не требует хранения данных об истории активности пользователя внутри системы, поскольку профиль пользователя «ВКонтакте» регулярно обновляется, а также решает проблему холодного старта за исключением случая, когда пользователь имеет пустой профиль.

Во избежание большого количества данных об объектах рекомендаций системы и введения своей системы классификаций объектов были использованы базы данных и системы классификаций сторонних сервисов, таких как `lastfm` и `themoviedb`.

ЛИТЕРАТУРА

1. Глибовец Н.Н. Использование JADE (Java Agent Development Environment) для разработки компьютерных систем поддержки дистанционного обучения агентного типа. URL: http://www.ebiblioteka.lt/resursai/Uzsienio%20leidiniai/IEEE/Russian/2005/Nr%203/OTO_2005_3_04.pdf
2. Журавский В. Простой поисковый робот. URL: <http://juravskiy.ru/?p=1005>
3. Охалкина Е.П., Лукоянов И.А., Воронов В.И., Воронова Л.И. Разработка и внедрение поискового робота для анализа интересов клиентов // Студенческий научный форум: VI Международная студенческая электронная научная конференция (15 февраля — 31 марта 2014 года). URL: <http://www.scienceforum.ru/2014/495/4758>
4. Пестряев А.А., Воронова Л.И. Мультиагентная система. Взаимодействие агента-собираателя с базой данных // Студенческий научный форум: VI Международная студенческая электронная научная конференция (Электронный ресурс), 15 февраля — 31 марта 2014 года. URL: <http://www.scienceforum.ru/2014/495/4036>
5. Ekstrand M.D., Riedl J.T., Konstan J.A. Collaborative Filtering Recommender Systems // Retrieved January 26, 2014, from Social Computing Research at the University of Minnesota website. URL: <http://files.grouplens.org/papers/FnT%20CF%20Recsys%20Survey.pdf>
6. Lane J. Musicians Find Fans At Grooveshark Artists. URL: <http://blogcritics.org/musicians-find-fans-at-grooveshark-artists/>
7. Ricci F., Rokach L., Shapira B. Introduction to Recommender Systems Handbook, Recommender Systems Handbook, Springer.
8. Sahebi S., Cohen W.W. Community-Based Recommendations: a Solution to the Cold Start Problem. URL: <http://d-scholarship.pitt.edu/13328>
9. Surfingbird. Официальный сайт компании. URL: <http://surfingbird.ru/about>

REFERENCES

1. Glibovets N.N. Using JADE (Java Agent Development Environment) for developing computer systems to support distance agent-type learning. URL: http://www.ebiblioteka.lt/resursai/Uzsienio%20leidiniai/IEEE/Russian/2005/Nr%203/OTO_2005_3_04.pdf
2. Zhuravsky B. Simple search robot. URL: <http://juravskiy.ru/?p=1005>
3. Okhapkina E.P., Lukoyanov I.A., Ravens V.I., Voronov L.I. Development and implementation of a web crawler to analyze the interests of clients // Student Scientific Forum: VI International Student Electronic Scientific Conference (February 15 — March 31, 2014). URL: <http://www.scienceforum.ru/2014/495/4758>
4. Pestryaev A.A., Voronov L.I. Multi-agent system. Interaction between gatherer agent and database // Student Scientific Forum: VI International Student Electronic Scientific Conference (February 15 — March 31, 2014). URL: <http://www.scienceforum.ru/2014/495/4036>
5. Ekstrand M.D., Riedl J.T., Konstan J.A. Collaborative Filtering Recommender Systems. URL: <http://files.grouplens.org/papers/FnT%20CF%20Recsys%20Survey.pdf>
6. Lane J. Musicians Find Fans At Grooveshark Artists. URL: <http://blogcritics.org/musicians-find-fans-at-grooveshark-artists>
7. Ricci F., Rokach L., Shapira B. Introduction to Recommender Systems Handbook, Recommender Systems Handbook, Springer.
8. Sahebi S., Cohen W.W. Community-Based Recommendations: a Solution to the Cold Start Problem. URL: <http://d-scholarship.pitt.edu/13328>
9. Surfingbird. Official web-site. URL: <http://surfingbird.ru/about>