СУ "Св. Климент Охридски"

Факултет по математика и информатика

ДИПЛОМНА РАБОТА

Тема :
Система за анализ на клиентско мнение в социалните мрежи базирана на архитектура ориентирана към събитията

Образователно-квалификационна степен: Магистър

Дипломант: Дако Христов Даков, факултетен № М-23842

Научен ръководител: гл. ас. д-р Калин Николов, ФМИ

София, 2014

Съдържание

Увод ... 4
Архитектура ориентирана към събитията ... 6
Събитие ... 6
Какво има в едно събитие? ... 6
Архитектура ориентирана към събитията ... 7
Обработка на събитията ... 7
Нива на архитектурата ориентирана към събития ... 8
Приложни компоненти на архитектура ориентирана към събития ... 9
Значение на социалните мрежи за бизнес организациите ... 12
Социални мрежи и социална медия ... 12
Петте "най-големи" социални мрежи ... 13
Проблеми и предизвикателства ... 14
Цел на дипломната работа ... 15
Добиване на данни от социалните мрежи ... 16
Извличане на знания от данни ... 17
Добиване на данни от социалните мрежи ... 18
Проблеми при добиването на данни от социалните мрежи ... 20
Анализ на общностите ... 20
Анализ на отношението и откриване на мнения ... 21
Социални препоръки ... 22
Изграждане на модел на влиянието ... 23
Произход и разпространение на информацията ... 23
Сигурност, доверие и защита на личните данни ... 24
Съхранение на данните ... 25
Визуализация на информацията ... 25
Визуализация на информацията върху карта. Визуализация на данните в реално време ... 26
Разработка на приложение, което демонстрира описаните възможности на системата ... 27
Избор на програмен език и технологии за разработка на системата ... 28
Описание на използваните компоненти и технологии ... 29
Apache Camel ... 29
Apache ActiveMQ ... 31
Eclipse Jetty ... 31
Lingpipe ... 32
JMapViewer ... 33
Google Maps ... 33
Autolinker.js ... 34
Java Message Service ... 34
JavaScript Object Notation (JSON) ... 35
WebSocket ... 36
Описание на модулите на системата ... 36
Модул за връзка със външни системи и анализ на събитията ... 36
Канал за пренос на събитията ... 38

Модул за запис на събитията 38
Модул за възпроизвеждане на записани събития 39
Модул за визуализация на събитията 40
Уеб сървър за визуализация на събития 42
Описание на нормалния процес на работа на системата 43
Заключение – анализ на резултатите 48
Бъдещи подобрения: 48
Използвана литература: 50

Увод

През последните години се наблюдава рязко увеличение на потребителите на социалните мрежи, като през 2013 година, 73 % от хората имащи достъп до интернет използват поне една социална мрежа. Потребителите използват социалните мрежи за развлечение, но също така и за да споделят свои снимки, преживявания и мнения. В същото време бързото развитие на мобилните технологии позволява достъп до интернет не само от настолни компютри, но и от мобилни устройства. Хората ги използват за всичко, включително и за достъп до социалните мрежи. В края на 2013 година, 73 % от притежателите на смартфони използват приложения за достъп до социалните мрежи, което им позволява да взаимодействат с тях по всяко време на денонощието, независимо къде се намират.

В резултат на това развитие на технологиите социалните мрежи всеки ден получават милиони събития – мнения, снимки, разкази за преживявания и т.н.

От друга страна, все повече компании започват да използват маркетинговата сила на социалните мрежи – 97 % от големите компании използват социалните мрежи като част от своята маркетингова стратегия – компаниите наемат място за реклама в социалните мрежи, организират игри и състезания, с които се опитват да представят своите продукти на крайните потребители.

За компаниите е важна обратната връзка от потребителите относно техните продукти и услуги, особено за новите им продукти – удобни ли са новия модел маратонки на фирмата X, достатъчно сладък ли е новият сладоледъ Y, доволни ли са гостите от обслужването и условията във веригата хотел Z. И на по-високо ниво: как се възприема дадена търговска марка, каква е репутацията на дадена компания – въпроси, чиито отговори позволяват на компаниите да направят важни решения за управлението на своите продукти и услуги.

За да получат отговор на тези въпроси, компаниите трябва да анализират всички събития изпратени от потребителите на социалните мрежи всеки ден. Тези събитията трябва да бъдат разгледани поотделно, от тях да бъдат избрани тези, които се отнасят за определен продукт или услуга и да бъде оценено мнението на потребителя.

Изпълнението на подобна задача е невъзможно без наличието на автоматична система, която да обработи милионите събития, които потребителите на социалните мрежи изпращат ежедневно.

В тази дипломна работа ще разгледаме по-подробно значението на социалните мрежи за съвременния бизнес и съществуващите методи за добив и анализ на данни от социалните мрежи. След това ще демонстрираме как може да бъде създадена автоматична система за анализ на потребителското мнение, базирана на архитектура ориентирана към събитията.

Архитектура ориентирана към събитията

Събитие

Събитие е нещо забележително което се случва вътре или извън нашата система или предприятие. Едно събитие (работно или системно) може да означава проблем или предстоящо проблем, възможност, достигната граница или отклонение.

Дефиниция и проявление - терминът събитие често се използва взаимнозаменяемо както за спецификацията (определение) на вид събитие, така и за всяка отделна проява на събитието.

За да има събитието смисъл за всички потребители, както хора, така и машини, е важно събитието да бъде дефинирано с бизнес терминологията, а не като данни и работни единици.

Какво има в едно събитие?

Всяко отделно събитие има заглавие и тяло (header and body). Заглавието съдържа елементи, които описват конкретното събитие, като например специфичен индикатор, типа на събитието, име на събитието, времеви индикатор, за кой път се случва това събитие, кой го е създал или от къде идва. Тези елементи са едни и същи за всички събития от даден тип и се съдържат в спецификацията на събитието.

Тялото на събитието съдържа информация за това, което всъщност се е случило. Например ако в една търговска система сме дефинирали събитие за ниска наличност, тялото на събитието ще съдържа информация наличността на кой продукт е паднала под дадена граница.

Тялото на събитието трябва да бъде добре описано, за да може всяка заинтересувана страна да използва събитието, без да се налага да изисква допълнителна информация от първоизточника. В събитието за ниска наличност например, тялото на събитието трябва да съдържа не само идентификатор за продукта, но и описание за самия продукт, каква е наличността в момента и какви са зададените граници за наличност. За да се осигури правилното разбиране на събитията от всички участващи страни е добре да се използва бизнес лексикон или онтология.

Архитектура ориентирана към събитията

При архитектурата ориентирана към събитията, нещо забележително се случва вътре или извън рамките на нашия бизнес и знанието за това събитие се разпространява до всички заинтересовани страни (човешки и машинни). Всяка страна обработва и оценява събитието и ако е необходимо извършва някакво действие. Това действие може да бъде извикването на дадена услуга, започването на бизнес процес или създаването и публикуването на друго събитие.

По своята същност, архитектурата ориентирана към събития е с много ниска свързаност и с висока разпределеност. Създателят/източникът на събитието само знае, че то е публикувано. Той не знае и не се интересува от това кой е получил събитието и как го е обработил. Като резултат от това, дадено събитие много трудно може да се проследи в рамките на една система с много участници. Затова и се препоръчва архитектури ориентирани към събитията да бъда използвани за асинхронна обработка на данни и информация. [1]

Обработка на събитията

Съществуват три основни стила на обработка на събитията: прост, поточен (непрекъснат) и сложен (комплексен). В една добре развита архитектура ориентирана към събитията трите стила могат да се използват заедно.

При простата обработка на събития, когато се появи важно събитие, се предприема някакво определено действие. Този стил се използва най-често в системи за реално време, като целта е да се намали времето за реакция, а оттам и разходите за предприятието.

При поточното обработване на събития се наблюдават както по-важни, така и "обикновени" събития. При този стил на работа всяко събитие се обработва, като се оценява неговата значимост и се предава на заинтересованите участници. Поточното обработване на събитията се използва за разпределянето на информацията във предприятието, което спомага за своевременното взимане на правилни решения.

Сложното обработване на събития (Complex Event Processing) се свежда до обработването на група събития и предприемане на определени действия. Събитията, както "обикновени" така и "по-важни", могат да бъдат от различен тип и да са се случили през дълги интервали от време. Връзката между отделните събития може да

бъде причинна, времева или пространствена. Сложната обработка на събития налага използването на по-сложни тълкувания на събитията, сложни системи за намиране на тенденции и връзки. Този стил на обработка на събитията се използва за разпознаването на аномалии, заплахи и потенциални възможности.

Нива на архитектурата ориентирана към събития

Архитектурата ориентирана към събития може да бъде разделена на четири логически нива:

Генератори на събития: всяко събитие има източник. Този източник може да бъде дадено приложение, хранилище за данни, услуга, бизнес процес, предавател, сензор или система за съобщения (e-mail, приложения за обмен на кратки съобщения и др.). Едно просто събитие може да бъде обработено от входна система (рутер или филтър) и в резултат да бъде създадено ново "важно" събитие.

Тъй като има различни генератори на събития, някои от събитията може да бъдат в непознат или некоректен формат, който не може да бъде обработен от всички участници в архитектурата. В такъв случай се налага използването на трансформатор, който да преведе събитието в стандартен вид, преди то да бъде допуснато до канала на събитията.

Канал на събитията: Канала на събитията най-често е система за пренос на съобщения. Тази система се използва за транспортиране на стандартно форматираните събития между генераторите на събития, системите за обработка на събитията и други абонати.

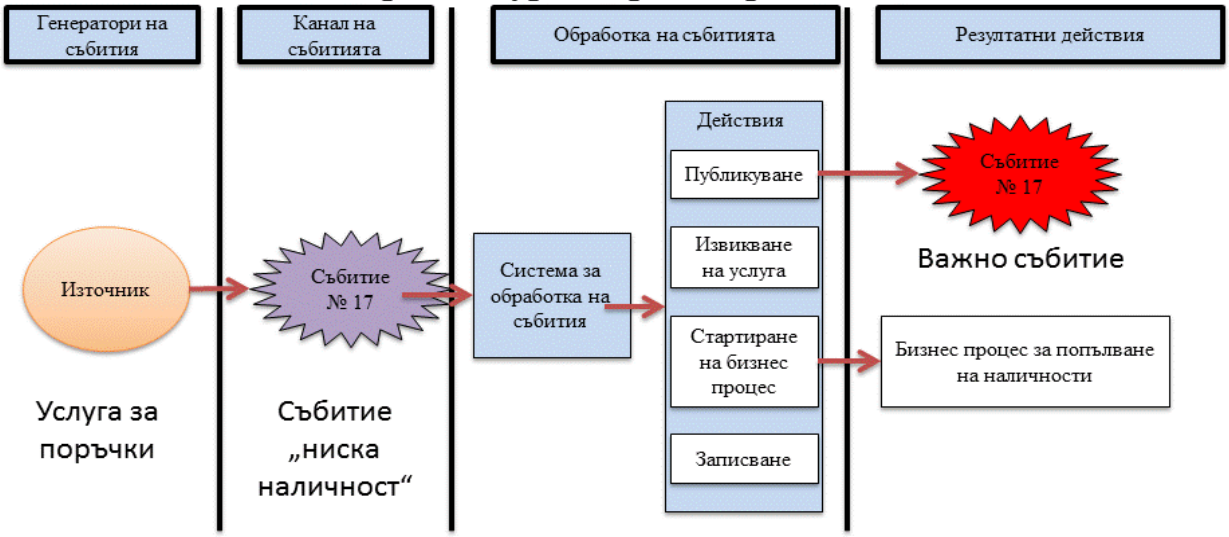
Обработка на събитията: На това ниво всяко получено събитие се оценява спрямо зададени правила и се предприемат съответни действия. Правилата за обработка на събитията се определят от нуждите на заинтересуваните страни и не са съгласувани със генераторите на събитията.

Действията могат да бъдат: използването на услуга, стартирането на бизнес процес, публикуване на събитието в други системи, директно известяване на хора или машини, генериране на ново събитие и/или записването на събитието с архивна цел.

Събитията се обработват от системи за обработка на събития (Event Processing Engines). Простите системи за обработка на събитията разглеждат всяка събитие поотделно. Комплексните системи за обработка на събитията, обработват всяко събитие в контекста на миналите и евентуални бъдещи събития.

Резултатни действия: Едно или няколко свързани събития, могат да предизвикат множество резултатни (последващи) действия. Действието може да предизвика употребата на услуга или започването на бизнес процес, както и известяването на абонати. Абонатите могат да бъдат хора, приложения, вече активни бизнес процеси, хранилища за данни или други автоматични агенти. Резултатните събития трябва да бъдат публикувани в стандартния формат. Ако някой от абонатите има нужда от специален формат, той най-често трябва да се обърне към интеграционна система.

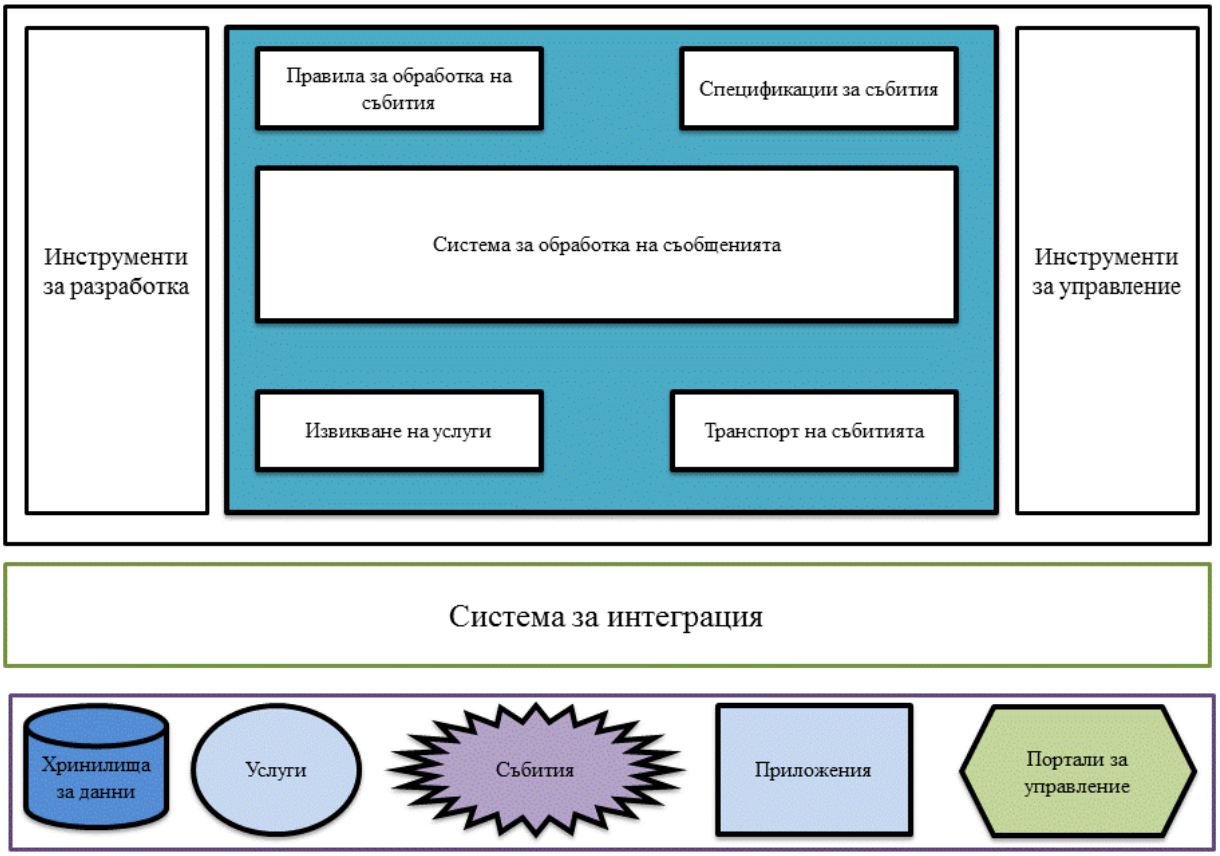
Нива на архитектурата ориентирана към събития



Приложни компоненти на архитектура ориентирана към събития

До сега разгледахме някои от приложните компоненти на архитектурата ориентирана към събития. Както е показано на следващата фигура, те могат да бъдат разделени в пет категории:

Приложни компоненти на архитектура ориентирана към събития



Метаданни за събитията: Една добра архитектура ориентирана към събитията трябва да има структура на метаданните. Метаданните включват спецификациите на различните типове събития, както и правилата за тяхното обработване. Спецификациите на събитията трябва да бъдат достъпни за генераторите на събития, трансформаторите, системите за обработка и абонатите.

Обработка на събитията: В тази категория спадат системите за обработка на събитията и самите единични събития. Простите системи за обработка, често са разбработени от самите им потребители или отделни разработчици. Комплексните системи за обработка на данни, най-често се доставят от големите софтуерни компании. В момента такива системи предлагат компаниите EsperTech, IBM, Informatica, Oracle, Progress/Apama, Software AG, Starview Technology, StreamBase, Sybase (SAP) и Tibco. Отделните събития най-често се запазват и архивират с цел одит и анализ на тенденции.

Инструменти за разработка: Необходими са различни инструменти за дефиниране на типовете събития, състяване правилата за обработка и управление на абонаментите. Инструментите за управелние предоставят възможност за управление и

наблюдение на системите за обработка на събитията, потока на събития, както и статистика за генерираните, транспортирани и обработени събития.

Интеграция в предприятието: Системите за интеграция играят важна роля в архитектурата ориентирана към събития. Интеграционните нужди включват: предварителна обработка на събитията (филтрация, пренасочване и трансформация), транспортиране на събития, извикване на услуги, стратиране на бизнес процеси, достъп до информация от системите на предприятието.

Източници и консуматори: Това са ресурсите в едно предприятие – приложения, услуги, бизнес процеси, хранилища за данни, хора, автоматични агенти и системи. Всички те могат да генерират събития или да предприемат действия, предизвикани от дадено събитие.

Конфигурацията на тези компоненти е различна за различните предприятия, в зависимост от обема и обработкана на събитията, интеграционните системи и разположението на източниците и консуматорите.

Значение на социалните мрежи за бизнес организациите

Социални мрежи и социална медия

Въпреки че са сравнително нов феномен, социалните мрежи се превръщат във все по-важна част от маркетинговата платформа на големия бизнес. Социалните мрежи и зараждащата се в тях социална медия предоставят нови средства на предприемачите да достигат и взаимодействат със своите клиенти. Докато в началото, социалните мрежи и медия се считаха за временно увлечение, все повече и повече бизнеса им обръща внимание и се интересува от тях. Освен това с течение на времето социалните сайтове разработиха различни инструменти, с които дадоха достъп на бизнеса до своите потребители. Ето и няколко факта за взаимодействието между социалните мрежи и бизнеса:

- 97 % от големите компании използват социалните медии, като част от своята маркетингова платформа, като
- 62 % от служителите в отдел маркетинг прекарват повече от един ден седмично, отделен само за социалните мрежи
- 89 % от компаниите, който обръщат специално внимание на социалните медии съобщават, че правят по-големи продажби благодарение на това.
- 43 % от хората на възраст от 20 до 29 години, прекарват над 10 часа седмично в социалните мрежи.
- Потребителите на социални мрежи прекарват в различните социални сайтове на 2 милиарда часа месечно. [2,8]

Компаниите, отделящи място на социалните мрежи в своите маркетинг кампании, получават следните предимства:

- по-голяма известност на марката/компанията сред широката публика
- увеличен трафик към корпоративните им интернет сайтове
- по-голям брой "лоялни" клиенти
- по-добър ранг в интернет търсачките
- редуциране на разходите за маркетинг
- увеличение на продажбите
- образуването на нови бизнес партньорства [6]

Петте "най-големи" социални мрежи

Бизнеса използва основно пет големи социални мрежи. Ето кратко описание на всяка от тях:

1. Facebook – въпреки че започва като сайт за комуникация между студенти, чиято основна цел е споделянето на снимки, Facebook се разраства и се превръща в най-голямата социална мрежа в света. С течение на времето се разработват и редица инструменти, които са в помощ на бизнеса и дават достъп до потребителите на мрежата. Основния начин, по който бизнеса използва Facebook, е чрез създаването на "страница" за своята марка, която потребителите могат да започнат да следят. Страниците могат да бъдат използвани за маркетинг на продукти, огласяването на специални оферти или промоции и изграждането на самата търговска марка. Понастоящем Facebook има 1.28 милиарда активни потребители (данните са от март 2014).
2. Twitter – също като Facebook, Twitter започва като социален инструмент за комуникация, позволяващ на приятели да поддържат връзка. И също като Facebook, с течение на времето се превръща във напълно развита социална платформа със маркетингови инструменти. Комуникацията в Twitter се осъществява посредством 140 символни tweet-ове (къси съобщения), които се появяват на личната страница на всеки потребител, който следи дадена регистрация. Използването на хештагове (дума или фраза, предшествана от #) позволява на потребителите да търсят и групират съобщения по различни теми, като ги обозначават със съответните хештагове. Twitter има над 255 милиона активни потребители (данните са от април 2014), които изпращат средно по 500 милиона съобщения дневно.
3. Google + е сравнително нова платформа, но бързо става популярна сред фирмите, които искат да придобият популярност в света на социалните мрежи. Google + предоставя възможности подобни на Facebook и Twitter, но ги допълва и с няколко бизнес ориентирани възможности: опцията "Promote" позволява допълнително конфигуриране на промоционалните съобщения според групата на която искате да ги изпратите. Освен това Google + предоставя достъп до допълнителни инструменти, с които може

да бъде следена ефективността на маркетинговата кампания и активността на потребителите по време на посещението на вашата страница. Допълнително от Google + са подготвили серия въвеждащи клипове, насочени специално към бизнес маркетинга. Мрежата има 540 милиона активни потребители (данните са от октомври 2013).

4. LinkedIn е социална мрежа ориентирана към работещите професионалисти. Функционираща от 2003 година, тя е насочена по-скоро към споделянето на професионална информация и изграждането на бизнес връзки, отколкото към забавно, лично или неформално съдържание. LinkedIn предлага на своите потребители възможността да попълнят своето резюме да търсят нови предложения за работа, както и да следят за определени позиции. За компаниите предлага възможност да публикуват промоционална информация, оповестяват свободни позиции за работа, както и таргетирана реклама в рамките на самата мрежа. LinkedIn има 300 милиона регистрирани потребители (данните са от април 2014).
5. YouTube – въпреки, че възниква като платформа за споделяне на видео клипове, напоследък YouTube, започва да придобива все повече чертите на социална мрежа – регистрираните потребители имат възможност да създават свои собствени канали и колекции от видео клипове, както и да следят каналите на други потребители. Тъй като YouTube е собственост на Google, тя е особено привлекателна за бизнес компаниите, поради интеграцията си с маркетинг инструментите на Google. YouTube има над 800 милиона уникални потребители всеки месец, за които не е необходимо да бъдат регистрирани за да достъпват съдържанието на сайта. 65 % компаниите, които провеждат маркетинг във социалните мрежи искат да използват YouTube за своите кампании, поради огромния ефект на видео клиповете и лесния достъп до клиенти от цял свят. [7]

Проблеми и предизвикателства

Тъй като маркетинга в социалните мрежи е сравнително нова област, все още съществуват проблеми за успешното му използване от бизнеса:

- Въпреки, че почти всички компании, които са използвали маркетинг в социалните мрежи съобщават, че са успели да повишат продажбите и като цяло имат по-голяма популярност, все още не са разработени точни методи за измерване влиянието на социалните мрежи в рекламата.
- Има липсата на експерти по маркетинг в социалните медии
- Социалните мрежи дават директен достъп до потребители, но и потребителите имат директен достъп до компанията – фирмите трябва да се подготвени да отговарят на негативни коментари и забележки от страна на потребителите на социални мрежи, както и да предприемат своевременни действия за подобряване качеството на продуктите, в следствие на обратната връзка получена от социалните мрежи.

Цел на дипломната работа

Целта на тази дипломна работа е да се разработи система, която наблюдава социалните мрежи и събира, анализира и визуализира данни за клиентското мнение за предварително зададена марка или продукт. Тази система ще бъде базирана на архитектура ориентира към събитията и ще се състои от няколко модула – връзка с външни системи и анализ на събития, получаване и обработване на събития, запис и възпроизвеждане и визуализация на събития.

Системата ще позволява на потребителя да преглежда получените събития в табличен вид, както и да получава информация на географска карта, за събитията, които съдържат геолокационни данни. Системата също ще позволява записа и възпроизвеждането на събития.

Добиване на данни от социалните мрежи

Широката употреба на социалните медии успя да генерира безпрецедентни количества социални данни. Социалните медии предоставят на своите потребители платформа, в която те лесно могат да споделят информация. Добиване на информация от данните в социалните медии може да извлече модели за действие, които могат да бъдат полезни на бизнеса и потребителите. Данните от социалните мрежи са огромни по своя обем, съдържат шум, неструктурирани са и са много динамични, което поставя нови предизвикателства пред специалистите.

Научноизследователска работа в областта на извличане на знания от данни е довела до разработването на редица методи, инструменти и алгоритми за обработването на огромни количества от данни, с цел разрешаването на всекидневни проблеми. Традиционното извличане на знания от данни се е превърнало в съставна част от много други науки, като например биоинформатиката, складирането на данни, бизнес анализа, прогнозиращия анализ, експертните системи и системите за подпомагане на решения. Целите на процеса на извличане на знания от данни е ефективното обработване на големи обеми от данни и извличането на смислени зависимости и разбиране същността на модела. Тъй като социалните медии са широко използвани за различни цели, всекидневно огромни количества от данни, произведени от потребителите са достъпни за обработка. Извличане на знания от социални данни може да подобри разбирането на нови социални феномени, да подобри бизнес анализа, с цел подобряване качеството на услугите и разработването на иновативни продукти и услуги. Например, извличането на знания от социални данни може спомогне в идентифицирането на влиятелните хора в огромни социални групи, да определи, кои са членовете на неформална или скрита група в даден социален сайт, да предугади настроението на клиентите и да помогне в проактивното планиране, да помогне в разработването на системи за препоръки за различни задачи – от предложения за закупуването на продукти, до предложения за нови приятелства, или да спомогне в подобряване на връзките между отделните потребители или потребителите и други организации.

Извличане на знания от данни

Извличане на знания от данни е процес, при който се обработват огромни количества от данни с цел откриването на нови знания, които могат да доведат до адекватни действия. Под извличане на знания може също така да се разбира процеса на извличане на полезна информация от необработени данни. Процесът най-общо се състои от следните задачи: предварително обработване на данните (data preprocessing), обработка на данните (data mining) и последваща обработка (post-processing). Тези задачи могат да бъдат изпълнявани, както поотделно, така и заедно. Извличането на знания от данни е съществена част от други научни области, като статистика, машинно самообучение, разпознаване на зависимости, бази от данни, визуализация, складиране на данни и откриване на информация. [3,4]

Алгоритмите за извличане на знания, се разделят на три основни групи: контролирани, неконтролирани и полуконтролирани обучаващи се алгоритми. Класифицирането е типичен пример за алгоритмите от контролираната група. При тях, обикновено данните са разделени на две части: тренировъчна и тестова част със предварително зададени определители (етикети, class labels). Контролирания алгоритъм изгражда класификационен модел от тренировъчните данни и го използва за да прави предвиждания. За да се оцени ефективността на алгоритъма, модела се прилага върху тестовите данни и се оценява точността на предвижданията. Типични примери за контролирани обучаващи се алгоритми са намиране на дърво на решенията, наивната бейсова класификация и метода на опорните вектори.

Неконтролираните обучаващи се алгоритми са предназначени за некласифицирани данни (данни без класов определител). Най-добре познатата група задачи от тази група алгоритми са тези за клъстеризация. За дадена задача, неконтролираните алгоритми изграждат модел, базиран на приликите или разликите между отделните обекти. Приликите или разлики между отделните обекти могат да бъдат изчислени по различни методи – чрез евклидово разстояние, разстояние на Минковски или разстояние на Махаланобис. Съществуват и други измерители на сходство, които могат да бъдат използвани – прост коефициент на сходство, коефициент на Жакар, косинусово сходство и коефициент за линейна корелация. Типични примери за неконтролирани обучаващи се алгоритми са алгоритмите за разделяне на к-средни клъстери, агломеративна и разделяща йерархична клъстеризация и клъстеризация основана на плътността.

Полуконтролираните обучаващи се алгоритми са най-подходящи, когато са на лице малко на брой класифицирани (класово определени) данни и голям брой некласифицирани обекти. Два типични примера за алгоритми от тази група са полуконтролираната класификация и полуконтролираното клъстеризиране. Първият използва класифицираните данни, за да класифицира неопределените обекти и по-този начин да определи по-точно границите на класификацията, а втория използва класифицираните данните, за да направлява клъстеризирането. Активните полуконтролирани обучаващи се алгоритми позволяват на потребителите да играят активна роля в процеса на обучение. В общия случай, потребителите са експерти в рамките на дадена област и техните знания се използват за категоризиране на определени обекти, за които алгоритъма вече е изградил класификация. Два от най-популярните алгоритми са Minimum marginal hyperplane и алгоритъма с максимално любопитство (maximum curiosity).

Извличане на знания от данни може да включва и други задачи като например търсенето на асоциативни правила, откриването на аномалии, отличителни черти, откриването на отделни индивиди или обекти и визуален анализ.

Добиване на данни от социалните мрежи

Потребителите на социалните мрежи генерират огромни количества данни всеки ден. Тази тенденция най-вероятно ще продължи за в бъдеще и количеството съдържание ще се увеличи експоненциално. Затова е изключително важно производителите, консуматорите и доставчиците на услуги да намерят начин да управляват и използват този поток от данни. Растежа на социалните медии се управлява от следните три въпроса:

- Как може всеки потребител де бъде чут?
- Кой е най-добрия източник на информация за даден потребител?
- Как може да се подобри преживяването на потребителя?

Отговорите на тези въпроси се крият във данните от социалните медии. Тези въпроси също така предоставят възможност на експертите в областта на извличане на информация от данни за разработването на нови алгоритми и инструменти.

Данните идващи от социалните медии, са различни от обикновените атрибут-стойност данни, които традиционно се използват при извличане на знания от данни. Данните от социалните медии в по-голямата си част са данни, генерирани от потребителите на социалните сайтове. Тези данни са огромни по количество,

зашумени, разпръснати, неструктурирани и динамични. Тези характеристики на данните поставят нови предизвикателства при извличането на данни и изискват нови, по-ефективни техники и алгоритми. Facebook и Twitter съобщават че има съответно по над 150 милиона и 90 уникални посетители на месец, само от САЩ. Според сайта за видео клипове YouTube, на ден се гледат на 5 милиарда видео клипа и всяка минута нови 100 часа видео се добавят към сайта. В сайта за обмен на изображения Flickr има над 7 милиарда снимки. Свободната енциклопедия – Wikipedia – има над 32 милиона статии на различни теми, със над 500 милиона читатели месечно.

В зависимост от коя социална платформа идват, данните те могат да съдържат много шум или грешки. Премахването на такъв тип данни е изключително важно за ефективното извличане на знания от данните. Изследователите са забелязали, че спамърите произвеждат много повече данни от обикновените потребители. Данните от социалните мрежи са разпръснати, тъй като няма единна организация, която да управлява всички социални сайтове. Разпръснатостта на данните представлява голяма проблем за учените, които искат да разберат потока на информацията в социалните медии. Също така, данните от социалните медии най-често са неструктурирани или имат коренно различна структура, което представлява още едно предизвикателство, за тези които искат да направят смислени наблюдения и изводи. Това се дължи на факта, че различните социални сайтове като LinkedIn, Facebook и Flickr имат различни цели и задоволяват различните нужди на своите потребители.

Сайтовете на социалните мрежи са много динамични и еволюират постоянно. Например Facebook, наскоро промени много концепции и добави нови особености на сайта като линия на времето за всеки потребител, възможността потребителите да създават свои скрити групи, както и редица промени свързани със сигурност и опазването на личните данни. Честите промени на потока и структурата на данните от социалните сайтове пречат на изследователите да разработят ефективни алгоритми за извличане на информация. Въпреки това съществуват още редица интересни въпроси свързани с човешкото поведение, които могат да бъдат изследвани с помощта на данните от социалните мрежи. Социалните мрежи могат да помогнат на рекламодателите да открият най-влиятелните хора и по-този начин да достигнат до възможно най-голям брой хора, като в същото време използват минимален бюджет. Социалните медии могат да помогнат на социолозите да открият особености на човешкото поведение, като например особености на поведение на индивида вътре и извън определени социални групи. Освен това, както видяхме наскоро социалните

медии играят жизнено важна роля в организирането и координацията на масови движения като например Арабската Пролет и Движението "Окупирай Уол Стрийт".

Проблеми при добиването на данни от социалните мрежи

Анализ на общностите

Общностите се състоят от групи от индивиди, които комуникират по-често с останалите индивиди вътре в общност, отколкото с такива извън нея. В зависимост от контекста общността може да бъде разглеждана като група, клъстер, сплотена подгрупа или модул. В социалните мрежи общностите могат да бъдат лесно наблюдавани, тъй като те позволяват лесното променяне на броя индивиди в общността. Социалните мрежи позволяват на хората да се свързват със свои приятели да намират други потребители със сходни интереси. Общностите в социалните мрежи, най-общо се разделят на две групи: ясно определени и скрити. Първите се формират по изричното желание на потребителите, докато вторите се зараждат от само себе си, в рамките на ежедневните взаимодействия. Анализа на общностите най-често се занимава със откриването, формирането и еволюцията на общностите.

Откриването на общности най-често състои в намирането на скрити общности в дадена социална мрежа. Методите за откриване на общности са разделени на четири категории:

1. Възлово ориентиран метод за откриване на общности – при него всеки възел трябва да отговаря на определени условия или да има определени свойства – общи атрибути, достижимост на възлите, честота на вътрешните и външните връзки и др.
2. Групово ориентиран метод за откриване на общности, при който една група трябва да притежава определени свойства, например минимална плътност.
3. Мрежово ориентиран метод за откриване на общности – при него групите се формират при разлагането на цялата мрежа на отделни по-малки части, типични примери са спектралната клъстеризация и модуларната максимизация.
4. Йерархично ориентиран метод за откриване на общности, при който целта е в цялата мрежа да се построи йерархия, която включва всички

общности. Този метод позволява разглеждането на дадената социална мрежа в различни мащаби. Представители на този тип методи са разделителната и агломеративната клъстеризация.

Социалните мрежи са много динамични. Общностите могат да се разширяват, свиват и разпадат за много кратък интервал от време. Методите за анализ на еволюцията на общностите цели да открие модели и зависимости в поведението на общностите в по-големи интервали от време и по-голям брой мрежови взаимодействия. Установено е, че колкото повече познати имам даден индивид в една група, толкова по-вероятно е той да се присъедини към нея, както и това, че групи, базирани на по-широк кръг от интереси растат по-бързо от високо специализираните групи.

Анализ на отношението и откриване на мнения

Анализ на отношението и откриване на мнения имат за цел автоматичното откриване на мнения и оценки споделени в съдържанието, генерирано от потребителите. Такъв тип анализ и инструменти позволяват на бизнес организациите да разберат какво е отношението към техните продукти, приемането на различните марки, отношението към новите продукти и им позволяват да управляват репутацията на своите компании. От друга страна тези инструменти също позволяват на потребителите да разберат, какво е глобалното отношение към дадена продукт или марка. Някой от социалните мрежи са разработили свои собствени инструменти и предоставят подробни справки за отношението към даден продукт в различни формати. Постоянното наблюдаване на отношението на потребителите към дадена компания или продукт в социалните медии е изцяло ново предизвикателство.

Анализа на отношението е труден, тъй като съдържанието на данните, генерирани от потребителите може да бъде двусмислено и неясно. Анализа включва няколко стъпки:

1. откриване на правилните документи
2. откриване на правилните части от документа, свързани с продукта, който ни интересува
3. определяне на цялостното отношение / мнение
4. анализиране на отношението и поставяне на количествена оценка

5. агрегиране на откритата информация и даване на цялостна оценка за клиентско отношение [5]

Основните компоненти на едно потребителско мнение са:

1. обекта, към който е изразено конкретното мнение
2. самото мнение за обекта
3. кой е автора на мнението

Обектите в общия случай се представят като съвкупност от отличителни черти, като всяка отличителна черта може да бъде изразена чрез краен брой синонимни думи и фрази. Анализа на мнението може да бъде извършен на различни нива: на база на целия документ, на отделни изречения или част от изречение или словосъчетание. Въпреки, че вече са поставени основите, анализа на отношение във сравнителни изречение все още остава трудна задача. Друг проблем при този вид анализ е липсата на "твърда земя" – система, по която да бъде оценена ефективността на приложения алгоритъм.

Социални препоръки

Традиционните системи за препоръки се опитват да предложат продукти на потребителите на база на агрегирани оценки на продукти и анализирайки списъците от покупки на предходните потребители. Системите за социални препоръки се възползват от данните придобити за даден потребител от социалните мрежи и ги агрегират с данните от традиционните системи. Системите за социални препоръки се основават на хипотезата, че хората, които са свързани в социалните мрежи имат сходни интереси (хомофилия), както и че потребителите могат лесно да бъдат убедени от по-влиятелните си приятели да купят даден продукт. Препоръките идващи от познат човек, са много по-ценни от произволно генерираните препоръки, които предоставят традиционните системи. Целта на системите за социални препоръки са да подобрят качеството на препоръките, но също така и да намалят броя на препоръчаните продукти, за да не се достига до информационно претоварване. Типични примери за такъв вид препоръки са препоръките за книги, базирани на списъците с прочетени книги на ваши приятели във Amazon или продуктите харесани от ваши приятели във Facebook или Twitter.

Изграждане на модел на влиянието

От доста време учените изучават моделите на влияние и хомофилията във социалните мрежи. От голямо значение е да се разбере дали дена социална мрежа се ръководи от влиянието на определени потребители и ли като цяла от хомофилия. Това е от значение за рекламна индустрия, например, защото ако дадена социална мрежа се ръководи от влиянието на определени потребители, то рекламодатели би трябвало да идентифицират тези влиятелни потребители и да ги стимулират да рекламират даден продукт или услуга на останалите потребители на мрежата. В случай обаче, че хомофилията доминира в дадена мрежа, то тогава би било по-добре ако просто голям брой потребители бъдат стимулирани, за да повишат продажбите. Тъй като в повечето социални мрежи двата модела съществуват паралелно, все още е трудно да се разграничат двете групи. Вече има няколко теоретични работи, чиято цел е дадат ясен алгоритъм за разграничаване на двата модела. Също така съществуват алгоритми за определяне на оптималния брой първоначално стимулирани потребители, така че броя на достигнати крайни потребители да бъде максимален, в рамките на предварително зададен бюджет.

Произход и разпространение на информацията

Учените изследват произхода и разпространението на информацията и вече има създадени редица модели, като например независимия каскаден модел, праговия модел, модела податлив-заразен и модела податлив-заразен-възстановен. Учените прилагат тези модели за да анализират разпространението на слухове, компютърни вируси и дори разпространението на заразни болести. От гледна точка на социалните медии, двата най-важни въпроса са: как се разпространява информацията в рамките на социалната мрежа и кои фактори влияят на скоростта на разпространение; как може да определим кои са правдоподобни източници на информация в социалната мрежа. Първия въпрос, вече е добре изследван и има няколко установени модела за анализ. Вторият въпрос все още е обект на проучване, тъй като класическите методи за установяване първоизточника на информация и неговата достоверност, не са приложими в разпределените и динамични социални медии.

Сигурност, доверие и защита на личните данни

Лесния достъп и все по-разпространената употреба на социалните мрежи, доведе до проблеми свързани със сигурността и защитата на личните данни. Възникват нови предизвикателства свързани с противоречивите нужди на потребителите – от една страна, потребителите искат да имат възможно най-много приятели и да споделят възможно най-много информация за себе си, но от друга страна всеки потребител би искал да има възможност да защити личната си информация, когато това се налага. Но за да бъдем общителни ние трябва да сме отворени и прозрачни, а за да бъдат данните ни защитени, трябва да ограничим споделянето. От другата страна социалните медии имат нужда да окуражават потребителите да бъдат открити, да се свързват с възможно най-много други потребители. В същото време социалните мрежи са длъжни да отблъскват всички атаки по сигурността на своите потребители и организации, които ги използват. С различната информация, която потребители споделят в социалните мрежи – както лична, така и информация за другите потребители, се излагат на риск от редица заплахи. Все по-често социалните мрежи са обект на активни и пасивни атаки – следене и дебнене, кибер тормоз, лъжлива реклама, спам, социално инженерство и пренасочване към сайтове, които копират на външен вид социалните мрежи.

Според няколко проучвания, само много малка част от потребителите сменят настройките за сигурност, когато се регистрират в нова социална мрежа. От друга страна по-голяма част от мрежите, по подразбиране, предлагат почти цялата информация за нови потребители на всеки, който се интересува от нея, с цел разширяване на потребителските кръгове. Когато всички данни са публично достъпни хакерите и други злонамерени потребители могат да ги използват за лични цели. Понякога злонамерени потребители могат да причинят физически и емоционален стрес. Има други проучвания, че дори даден потребител да е променил настройките си за лични информация, той пак може да бъде изложен на риск, ако другите потребители от групите, в които потребителят членува не променят своите настройки и не са внимателни.

Според редица проучвания, по-голяма част от потребителите на социалните мрежи не обръщат внимание на собствената си сигурност и споделят огромни количества информация за себе си и своите интереси – информация, която може да бъде използвана от злонамерени потребители. Някои от социалните мрежи започнаха да провеждат серии от инициативи по обучение на потребители по въпросите на

сигурност. Други на свои ред въведоха по-строги механизми за споделяне и достъп на информация.

Доверието в социалните мрежи, зависи от много фактори и не може лесно да бъде описано с модел в една информационна система. Учените са разработили няколко дефиниции за доверие в социалните мрежи, но нито една не е приета единодушно. Според наблюденията, доверието между двама души зависи от много фактори, като общи преживявания, изразените мнения и предприетите действия, склонността да се споделя нова информация, влиянието на други потребители и др. Друг важен аспект на доверието в социалните мрежи е достоверността на информацията, генерирано от други потребители. Вече са разработени няколко системи за оценка на достоверността на информацията, които могат да бъдат приложени в социалните мрежи.

Съхранение на данните

Социалните медии генерират огромни количества от данни. В края на 2013 година от Facebook съобщиха, че вече притежават над 300 Petabyte-а данни, като всеки ден потребителите добавят по над 500 Terabyte-а данни. В зависимост от целта на системата или анализа, съхранението на такова огромно количества данни може да не бъде оправдано. Предварителната обработка и филтрация, през която преминават данните, преди да бъдат анализирани и съхранени е от изключително значение за намаляване обема на обработената информация. Количеството на данни, които трябва да бъдат съхранени зависи също и от периода, за който правим анализ – ако анализите са за кратки периоди от време – минути, часове или дни, то тогава количеството на съхранени данни намалява значително. Когато анализа се извършва за дълги периоди от време, има две възможност – или всички данни да бъдат съхранени, в хранилище за данни с голям капацитет или данните въобще да не бъдат съхранявани, а да бъдат изваждани от употреба, веднага след техния анализ.

Визуализация на информацията

Главната цел на визуализирането на информацията е да покаже данните ясно и ефективно, чрез графични средства. Това не означава, че визуализацията трябва да е досадна за да изпълнява целта си или да е прекалено сложна, за да изглежда красиво. За

да предават информацията ефективно, визуализациите трябва да са едновременно естетични и функционални и да показват ключовите аспекти на информацията по интуитивен и разбираем начин. Въпреки това много често се случва така, че създаваме великолепно изглеждащи визуализации, които обаче се провалят в основната си функция – да предават информация.

Съществуват различни видове за визуализация на данни, най-разпространените сред които са:

- таблици
- диаграми
- графики
- карти

Визуализация на информацията върху карта. Визуализация на данните в реално време

С развитието на мобилните технологии и все по-широкия достъп до Интернет, информационните системи имат достъп до постоянно увеличаващ се обем от данни. Крайните потребители вече имат възможността не само да предадат своето съобщение в реално време, но и автоматично с него да предадат и редица съпътстващи данни: географско разположение, отчитания на различни датчици, и др. Тези нови възможности позволяват визуализирането на данните в реално време – данните могат да бъдат визуализирани веднага, след като постъпят от крайните потребители и бъдат обработени.

Наличието на съпътстващи данни, и в частност на геолокационни данни, позволява на информационните системи да визуализират информацията и върху географски карти – по този начин наблюдаващите потребители имат възможност да видят, къде точно е възникнало дадено събитие и да вземат по-добро решение за реакция.

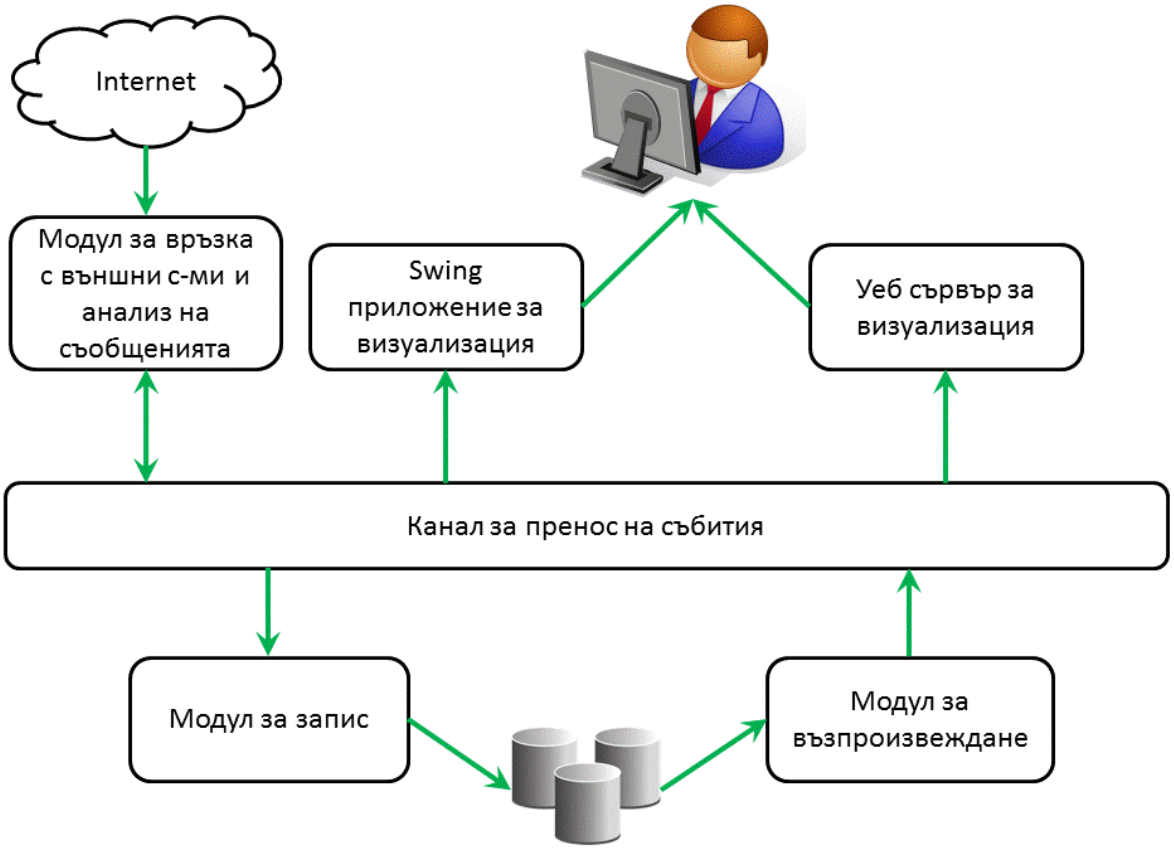
Разработка на приложение, което демонстрира описаните възможности на системата

За да бъдат изпълнените целите на дипломната работа бе разработена модулна система, базирана на архитектура ориентирана към събитията. Тя предоставя на своите потребители следните възможности:

- Осъществяване на връзка със социалните мрежи и абониране за получаване на нови събития възникнали в тях. Абонамента за събития може да бъде насочен към определена ключова дума или тема или просто да следи всички възникващи събития.
- Анализ на клиентското мнение – всяко събитие предизвикано от потребителско действие в социалните мрежи има някакво съдържание. Когато това съдържание е текстово, система може да извърши автоматичен анализ на съобщението и да го оцени като "положително", "неутрално" или "отрицателно".
- Системата предоставя възможност за запис и възпроизвеждане на прехванатите събития – всяко събитие постъпило в системата може да бъде съхранено. При нужда, системата може да възпроизведе вече записаните събития.
- За визуализиране на данните са разработени два модула за визуализация: Swing базирано приложение, което показва получените съобщение в табличен вид и върху географска карта.
- Уеб сървър, който съдържа уеб страница, която също показва получените съобщения в табличен вид и върху географска карта.

За конкретна демонстрация на системата ще изпълним следния сценарии: системата ще осъществява връзка със социалната мрежа Twitter и ще следи за потребителски съобщения насочени към компанията Starbucks. Когато такова съобщение бъде публикувано в социалната мрежа, системата ще го прихваща и обработва. След като съобщението бъде обработено, системата ще визуализира данните за съобщението във табличен вид, като в зависимост от клиентското мнение, всеки реда в таблицата ще бъде оцветен. Съобщенията, които съдържат геолокационни данни, ще бъдат визуализирани и върху географска карта. Системата включва следните компоненти:

- модул за връзка със външни системи и анализ на събитията
- канал за пренос на събитията
- модул за запис на събитията
- модул за възпроизвеждане на записани събития
- Swing приложение за визуализация на събитията
- Уеб сървър за визуализация на събитията



Избор на програмен език и технологии за разработка на системата

По-голяма част от системата е написана на програмния езика Java. Самия език, както и виртуалната машина върху която се изпълнява кода, предоставят широк набор от приложно програмни интерфейси и възможност за бърза и лесна интеграция между отделните модули.

В разработката на компонентите на уеб сървъра за визуализация се използвани езиците HTML и JavaScript.

След като системата получи събитие от социалните мрежи, тя го конвертира до JSON формат, заради неговата простата и лесна употреба – този формат може да бъде лесно използван както в Java частта на системата, така и в уеб сървърната част.

За интеграция между отделните модули на системата е използвана Java базираната софтуерна рамка за разработка Apache Camel. [9] Тя предоставя лесна интеграция между разнородни системи, осъществявайки комуникацията между тях на асинхронни съобщения. За пренос на тези съобщения използваме JMS базираната система Apache ActiveMQ.

В разработката на системата са използвани свободно разпространявани компоненти с отворен код: Apache Camel, Apache ActiveMQ и Eclipse Jetty, безплатната версия на библиотеката lingpipe за анализ на текста, както и безплатната библиотека JMapViewer от проекта OpenStreetMaps и JavaScript библиотеките Google Maps и Autolinker.js.

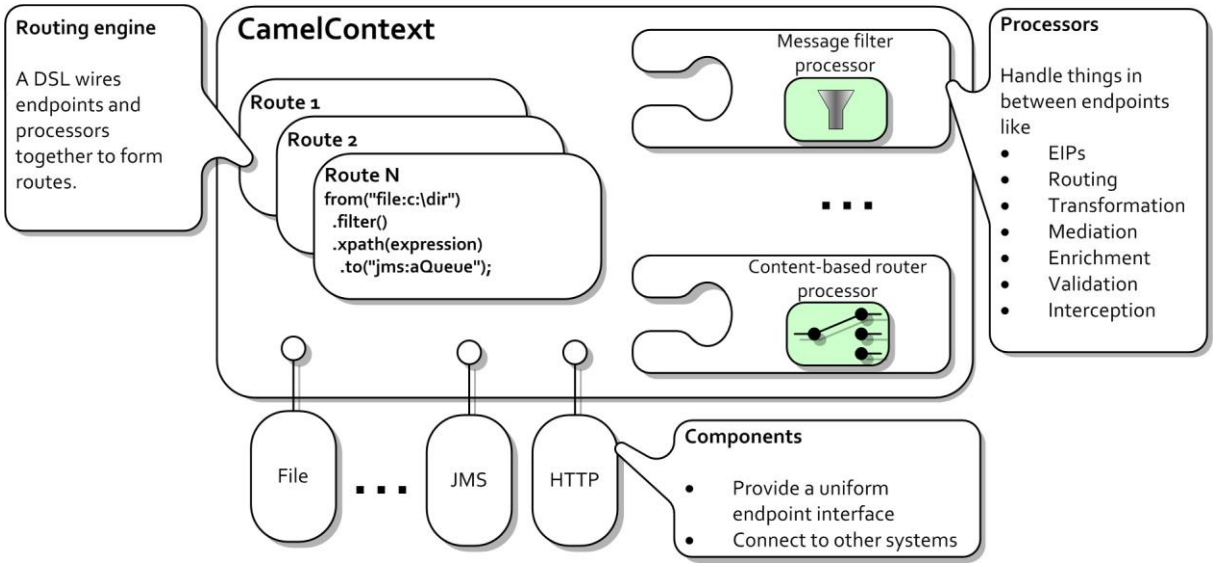
Описание на използваните компоненти и технологии

Apache Camel

Apache Camel е софтуерна рамка за разработка със отворен код, която е насочена към улесняването на интеграцията между различни софтуерни компоненти. Тя постига това като предоставя на разработчиците:

- готова разработка на всички широко използвани шаблони за интегриране (Enterprise Integration Patterns).
- възможност за връзка с различни видове транспортни протоколи и приложно-програмни интерфейси.
- лесен за употреба предметно-ориентиран език (Domain Specific Language), чрез който бързо и лесно могат да се навържат различните шаблони и транспорти.

За да използваме Apache Camel, е необходимо да се запознаем с няколко негови модула: компоненти (components), крайни точки (endpoints), процесори (processors) и предметно-ориентирания език, показани на следната фигура: [10]



Компонентите са механизма, чрез който се добавят нови функционалности към Apache Camel. Сърцевината на рамката е изградена от 13 основни компонента, а външната част на рамката съдържа други 80, които предоставят връзка с други системи. За да предоставят достъп до тези системи на други компоненти, всеки компонент предоставя своя крайна точка, която служи като интерфейс за комуникация. Като използва специфичните имена на тези крайни точки, потребителят получава достъп до различните системи по еднороден начин. Например за да получим съобщение от JMS опашка и да го запишем на файловата система използваме имена като "jms:aQueue" и "file:/tmp".

Процесорите се използват за манипулиране и предаване на съобщения между различните крайни точки. Всички интеграционни шаблони са дефинирани като отделни процесори или набор от процесори. Apache Camel съдържа над 40 вградени процесора, като също така предоставя възможност на потребителите да добавят свои собствени процесори да и ги включат в различни стадии от обработката на съобщението.

За навързването на отделните процесори и шаблони, Apache Camel дефинира предметно-ориентиран език, който е приложим в различните езици за програмиране, като например Java, Scala и Groovy. Също така позволява и правилата за обработка на съобщенията да бъдат дефинирани в XML конфигурационен файл.

Apache Camel съдържа и редица конвертори, които позволяват смяната на формата на съобщението в най-често използваните формати като например XML, CSV, JSON и други. [11]

Apache ActiveMQ

ActiveMQ е междинен интеграционен софтуер (Middleware) с отворен код. Той е ориентиран към преноса на съобщения и поддържа Java Message Service (JMS) 1.1 спецификацията. ActiveMQ предоставя пренос на съобщения с висока надеждност, бързодействие, мащабируемост и висока сигурност. ActiveMQ се разпространява под лиценза Apache 2.0, който позволява на разработчиците на софтуер да го доразвиват и разпространяват, без почти никакви ограничения.

Целта на ActiveMQ е да предостави интеграция между различни приложения, базирана на утвърдени стандарти и включваща възможно най-голям брой програмни езици и платформи. ActiveMQ се базира на JMS спецификацията, като към нея прибавя и редица други допълнения:

- Поддръжка на различни комуникационни протоколи: HTTP/S, IP multicast, SSL, STOMP, TCP, UDP, XMPP и др.
- Разнообразни механизми за сигурност и запазване на съобщенията – потребителите могат да използват вградените механизми или да добавят свои собствени.
- Интеграция със Java сървъри за приложения
- Различни клиентски приложно-програмни интерфейси за различни езици - C/C++, .NET, Perl, PHP, Python, Ruby и др.
- Брокер клъстъринг – няколко инстанции на ActiveMQ могат да работят заедно във федерирана мрежа с цел мащабируемост.
- Улеснено управление – ActiveMQ е ориентирано към всички потребители и не се нуждае от специално обучен администратор, за да работи ефективно.

Eclipse Jetty

Jetty е първоначално разработен като HTTP сървър компонент на сървъра Mort Bay през 1994 година. Следват няколко смени на собствениците на кода, докато през 2009 година кода Jetty не става част от компоненти стопанисвани от Eclipse фондацията. В момента Jetty се разпространява като приложение с отворен код под Apache и Eclipse лицензите.

Jetty е изцяло написан на Java и служи като уеб сървър и Servlet Container. Докато повечето уеб сървъри се използват за да доставят документи на хора, Jetty напоследък се използва повече за комуникация между отделни машини, обикновено в големи комерсиални мрежи. Причините за това са малките му системни и изисквания за памет и процесорно време, както и големия набор от поддържани протоколи. Jetty поддържа следните протоколи и стандарти:

- SPDY – експериментален протокол за ускорен пренос на данни по HTTP
- WebSocket – протокол за двупосочна комуникация през една TCP връзка
- OSGi – спецификация за модулна платформа
- Java Management Extension (JMX)
- Java Naming and Directory Interface (JNDI)
- Java Authentication and Authorization Service (JAAS)

Jetty се използва като контейнер за редица продукти: Apache ActiveMQ, Alfresco, Apache Geronimo, Apache Maven, Apache Spark, Google App Engine, Eclipse, FUSE, Zimbra, Lift, Eucalyptus, Red5, Hadoop и I2P.

Lingpipe

Lingpipe е набор от Java библиотеки, които има следните възможности:

- Преброяване броя на споменаванията на даден обект, например човек или марка.
- Свързване на споменати обекти със записи в база от данни.
- Откриване на връзка между обекти и събития.
- Класифициране на откъси текст по език, кодиране на знаците, жанр, тема или мнение.
- Поправяне на правописа в дадена контекст.
- Клъстеризация на документи по теми и откриване на тенденции с течение на времето.
- Разпознаване на частите на речта и разделяне на фрази.

JMapViewer

Безплатна библиотека JMapViewer се използва за изобразяване на географски карти в Java приложения за работни станции. JMapViewer работи с безплатните карти, предоставени от проекта Open Street Map (http://www.openstreetmap.org/).

Google Maps

Google Maps е картографска услуга и технология предоставена от Google за употреба на мобилни и работни станции. Тя предлага сателитни снимки, карти на населени места и "Street view" перспективата, която позволява на потребителите да направят виртуална разхода по улиците на някои населени места или интересни местности. Google Maps поддържа и функционалност за намиране на маршрут между две точки и планиране на времето в зависимост от използвания начин на предвижване – пеша, с автомобил, велосипед или обществен транспорт. Чрез отворения си приложно програмен интерфейс Google Maps позволява внедряването на карти в уеб страници. Сателитните снимки на Google Maps не се опресняват в реално време, но повечето изображения не са по-стари от три години. [12]

Google maps приложението за мобилни устройства е едно от най-често използваните приложения в световен мащаб, като през август 2013 над 54 % от всички притежатели на смартфони в света са използвали приложението поне веднъж.

Приложно програмния интерфейс – Google Maps API е достъпен от юни 2005 година и позволява внедряването на географски карти в приложения и уеб страници. Първоначално достъпен само през JavaScript, в последствие интерфейса е разширен да позволява достъп и чрез други технологии. По настояще услугата е безплатна, при условие че потребителите не генерират повече от 25000 карти на ден. Ако това ограничение бъде превишено, потребителя може да закупи платен лиценз и да продължи да използва услугата. В края на 2013 година, над 1 000 000 интернет сайта използват интерфейса на Google Maps, което прави този интерфейс един от най-използваните в света.

Autolinker.js

Autolinker.js е безплатна JavaScript библиотека разработена от Грег Джейкъбс (https://github.com/gregjacobs/Autolinker.js). Тя се използва за автоматично разпознаване на интернет връзки в неформатиран текст и превръщането им в използваеми HTML тагове. Библиотеката разпознава интернет връзки, със и без предшестващ протокол, e-mail адреси, както и Twitter псевдоними.

Java Message Service

Java Message Service (JMS) е приложно програмен софтуер, написан на програмния език Java, служещ като междинен интеграционен софтуер за комуникация между два или повече клиента, посредством съобщения. JMS е също така и стандарт, дефиниран във JMS спецификацията, позволяващ на Java базирани клиенти да създават, изпращат, получават и четат съобщения. Той позволява асинхронна комуникация между независимите компоненти на дистрибутирани системи. [13]

JMS има следните елементи:

- JMS Provider – доставчик – приложение, което изпълнява изискванията на JMS интерфейса за междинен интеграционен софтуер. Доставчиците са или Java приложения или адаптери към не-Java системи за междинна интеграция.
- JMS Client – клиент – приложение или процес, което изпраща или получава съобщения.
- JMS Producer/Publisher – производител/подател – клиент, който създава и изпраща съобщения.
- JMS Consumer/Subscriber – консуматор/абонат – клиент, който получава и чете съобщения.
- JMS Message – съобщение – обект, който съдържа данните предавани между два клиента.
- JMS Queue – опашка – временно хранилище за съобщения, който са били изпратени и трябва да бъдат получени и прочетени, само от един консуматор. Противно на името, редът на обработване на съобщенията не е задължително същия, в който съобщенията са постъпили в опашката.

Опашката гарантира единствено, че всяко съобщение ще бъде обработено само веднъж.

- JMS Topic – тема – механизъм за разпределяне на съобщения до множество абонати. Всяко съобщение изпратено до дадена тема се препраща до всички абонати.

JavaScript Object Notation (JSON)

JSON е формат със отворен стандарт, който използва четим текст за пренос на обекти от данни състоящи се от двойки ключ-стойност. Предимно се използва за пренос на данни между сървъри и уеб приложения, като алтернатива на Extensible Markup Language (XML). Въпреки че първоначално насочен към JavaScript, JSON е независим формат и методи за генериране и работа с него има във почти всички съвременни програмни езици. [14]

Въпреки, че JSON е широко използван, за него все още не съществува формално приета спецификация. Обектите в JSON поддържат следните полета: число, низ от символи, булеви променливи, масиви, празни стойности (null) и други обекти. Ето примерно описание на човек в JSON формат:

```
{
    "firstName": "John",
    "lastName": "Smith",
    "isAlive": true,
    "age": 25,s
    "height_cm": 167.64,
    "address": {
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": "10021-3100"C
    },
    "phoneNumbers": [
        { "type": "home", "number": "212 555-1234" },
        { "type": "fax",  "number": "646 555-4567" }
    ]
}
```

Съществува и проект на спецификация за JSON Schema, която подобно на XML Schema ще позволява валидацията на получените JSON обекти. За жалост проекта е в много ранен стадии и в различните езици за програмиране са създадени различни

механизми за валидация на JSON обекти, които не винаги работят добре с обекти създадени от други езици.

WebSocket

WebSocket е протокол за двупосочна комуникация през една TCP връзка. Протокола е стандартизиран през 2011 година и е създаден за връзка между уеб клиенти и уеб сървъри, въпреки че всяко клиент-сървър приложение би могло да го използва.

Благодарение на WebSocket протокола е възможно по-голямо взаимодействие между сървъра и уеб браузъра, което позволява постоянното опресняване на съдържанието и разработката на по-интерактивни приложения като например игри в реално време. Протокола постига това като предоставя стандартизиран начин на сървъра да изпраща данни на клиента, без той да ги е заявил изрично, като също така позволява двупосочната комуникация между двете странни, през постоянно отворена връзка. По този начин се осъществява двупосочна комуникация и обмен на данни между сървъра и браузъра.

Описание на модулите на системата

Модул за връзка със външни системи и анализ на събитията

Модула за връзка с външни системи и анализ на събитията представлява Java приложение, базирано на Apache Camel, което няма потребителски интерфейс. Неговата цел е да комуникира със социалната мрежа Twitter и да препраща получените събития към останалите модули в системата. Модула обработва входните събития и анализира текста на полученото съобщение. След като обработката приключи, модула изпраща JMS съобщение в JSON формат към вътрешната част на системата

Модула прави инстанция на Apache Camel със следната конфигурация:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.springframework.org/schema/beans
         http://www.springframework.org/schema/beans/spring-beans.xsd
         http://camel.apache.org/schema/spring http://camel.apache.org/schema/spring/camel-spring.xsd">

<bean id="activemq" class="org.apache.activemq.camel.component.ActiveMQComponent">
         <property name="brokerURL" value="tcp://localhost:61616"/>
</bean>
```

```
        <bean id="converter" class="events.gateway.Converter"/>

        <camelContext id="defaultContext" xmlns="http://camel.apache.org/schema/spring" autoStartup="true">

                <dataFormats>
                        <json id="json" library="Jackson"/>
                </dataFormats>

                <route >
                        <from uri="twitter://search/tweets?type=polling&amp;delay=5&amp;keywords=#starbucks&amp;
                                accessToken={{accessToken}}&amp;accessTokenSecret={{accessSecfret}}&amp;
                                consumerKey={{consumerKey}}&amp;consumerSecret={{consumerKey}}"/>
                        <process ref="converter" />
                        <marshal ref="json"/>
                        <to uri="activemq:topic:twitterEvents"/>
                </route>

        </camelContext>
</beans>
```

Горната конфигурация извършва следните операции:

1. Конфигурира JMS компонент, наречен "activemq", насочен към локалната машина и порт 61616.
2. Инстанциира класа "events.gateway.Converter" и му назначава псевдоним "converter".
3. Декларира формат на данни, според който конвертирането на данни в JSON формат ще се извършва от библиотеката "Jackson".
4. Дефинира маршрут, който на всеки 5 секунди търси съобщения в Twitter, които съдържат ключовата дума "#starbucks" и ги препраща, към класа converter. След като те бъдат обработени от класа converter, данните се конвертират в JSON формат и се препращат към JMS темата "twitterEvents".

Класа Converter преработва суровите съобщения от Twitter, като от тях изважда датата и часа на създаване на съобщението, потребителското име на автора, текста на самото съобщение и евентуално геолокационната информация, ако такава присъства в съобщението. След това текста бива анализиран с помощта на библиотеката lingpipe, при което му се поставя една от трите оценки: "положителен", "неутрален" или "отрицателен". Всички тези данни се записват като полета на един Java обект и се връщат на Camel за довършване на обработката – обекта се конвертира до JSON format и се препраща към JMS компонента.

Канал за пренос на събитията

Канала за пренос на данни представлява инстанция на Apache ActiveMQ. Той се използва като връзка между всички останали модули на системата. Модула за входна връзка изпраща съобщения към JMS темата "twitterEvents". Модула за запис на събития, модула за визуализация на събития и уеб сървъра за визуализация получават съобщения от тази тема. Модула за възпроизвеждане на записани събития изпраща съобщения към JMS темата "twitterEventsOld", от която четат само модула за визуализация и уеб сървъра за визуализация.

Инстанцията на ActiveMQ по подразбиране автоматично създава нова тема, когато към нея бъде изпратено съобщение или има нов абонат, затова не е нужно предварително конфигуриране. Потребителите могат да използват административната конзола, за да наблюдават потока от съобщения през системата или ако искат да направят настройки за преноса на данни – да променят настройките за сигурност, броят на препращания на съобщенията при грешка и т.н.

Модул за запис на събитията

Модула за запис на събитията представлява Java приложение, базирано на Apache Camel, което няма потребителски интерфейс. Неговата роля е да слуша за съобщения преминаващи през канала за пренос на данни и да ги записва на локалния диск.

Модула прави инстанция на Apache Camel със следната конфигурация:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.springframework.org/schema/beans
         http://www.springframework.org/schema/beans/spring-beans.xsd
         http://camel.apache.org/schema/spring http://camel.apache.org/schema/spring/camel-spring.xsd">

         <bean id="activemq" class="org.apache.activemq.camel.component.ActiveMQComponent">
                  <property name="brokerURL" value="tcp://localhost:61616"/>
         </bean>

         <camelContext id="defaultContext" xmlns="http://camel.apache.org/schema/spring" autoStartup="true">
                  <route >
                           <from uri="activemq:topic:twitterEvents"/>
                           <to uri="file://c:/twitterStorage"/>
                  </route>
         </camelContext>
</beans>
```

Горната конфигурация извършва следните операции:

1. Конфигурира JMS компонент, наречен "activemq", насочен към локалната машина и порт 61616.

2. Конфигурира Camel инстанция, в която дефинира маршрут, който насочва събитията от JMS темата "twitterEvents" към файловия компонент, който ги записва в директорията "c:/twitterStorage" на файловата система.

Всяко получено съобщение от модула за запис на събитията бива записано в отделен файл, чието име се състой от идентификатор на системата и уникален номер, назначен от Apache Camel.

Модул за възпроизвеждане на записани събития

Модула за възпроизвеждане на записани събития представлява Java приложение, базирано на Apache Camel, което няма потребителски интерфейс. Когато го стартираме, приложението изчита всички записани събития от директория на файловата система и ги изпраща до конфигурираната JMS тема.

Модула прави инстанция на Apache Camel със следната конфигурация:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.springframework.org/schema/beans
         http://www.springframework.org/schema/beans/spring-beans.xsd
         http://camel.apache.org/schema/spring http://camel.apache.org/schema/spring/camel-spring.xsd">

         <bean id="activemq" class="org.apache.activemq.camel.component.ActiveMQComponent">
                  <property name="brokerURL" value="tcp://localhost:61616"/>
         </bean>

         <camelContext id="defaultContext" xmlns="http://camel.apache.org/schema/spring" autoStartup="true">
                  <route >
                           <from uri="file://c:/twitterStorage?noop=true" />
                           <to uri="activemq:topic:twitterEventsOld"/>
                  </route>
         </camelContext>
</beans>
```

Горната конфигурация извършва следните операции:

1. Конфигурира JMS компонент, наречен "activemq", насочен към локалната машина и порт 61616.
2. Конфигурира Camel инстанция, в която дефинира маршрут, който инструктира файловия компонент да изчете всички записани събития от директорията "c:/twitterStorage" и без да ги променя или изтрива да ги предаде към JMS компонента, който ги изпраща към "TwitterEventsOld" темата.

При нормалното функциониране на системата, модула за възпроизвеждане на записани събития не функционира. Той се използва от потребителите, когато те искат да прегледат стари събития или след рестартиране на цялата система, за да бъде възстановено предишното състояние.

Модул за визуализация на събитията

Модула за визуализация на събитията представлява Java приложение с потребителски интерфейс, базирано на Apache Camel и използващо библиотеката JMapViewer.

Модула използва следните по-важни класове:

- DataEntry – клас, който съответства на едно получено събитие.
- JMapViewer – клас от библиотеката JMapViewer, който се използва за визуализиране на карти.
- TableModel – клас, който наследява Java класа AbstractTableModel и служи като модел за таблицата, която се показва на потребителя. Класа TableModel, съдържа множество инстанции на класа DataEntry и позволява работа с тях.
- TabbedFrame – клас, който наследява Java класа JFrame и служи като контейнер за графичната част на приложението. Той има два раздела – един за таблицата, в която се изобразяват получените съобщения и един за географската карта.
- ReceiverProcessor – клас, който служи като връзка между Apache Camel и останалата част от приложението. Всеки път когато бъде получено ново съобщение, Apache Camel извиква метода process() на класа, който обработва полученото съобщение : ако съобщените съдържа геолокационни данни, се поставя маркер с необходимия цвят в зависимост от оценката на текста върху географската карта; от останалите данни се прави инстанция на класа DataEntry и се добавя към модела на таблицата.

Модула прави инстанция на Apache Camel със следната конфигурация:

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

```xml
<beans xmlns="http://www.springframework.org/schema/beans"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.springframework.org/schema/beans
         http://www.springframework.org/schema/beans/spring-beans.xsd
         http://camel.apache.org/schema/spring http://camel.apache.org/schema/spring/camel-spring.xsd">

         <bean id="map" class="org.openstreetmap.gui.jmapviewer.JMapViewer" />
         <bean id="tableModel" class="events.gui.TableModel" />

         <bean id="tabbedFrame" class="events.gui.TabbedFrame" depends-on="map, tableModel">
                  <constructor-arg index="0" ref="tableModel" />
                  <constructor-arg index="1" ref="map" />
         </bean>

         <bean id="receiverProcessor" class="events.gui.ReceiverProcessor" depends-on="map, tableModel">
                  <constructor-arg index="0" ref="tableModel" />
                  <constructor-arg index="1" ref="map" />
         </bean>

         <bean id="activemq" class="org.apache.activemq.camel.component.ActiveMQComponent">
                  <property name="brokerURL" value="tcp://localhost:61616" />
         </bean>

         <camelContext id="defaultContext" xmlns="http://camel.apache.org/schema/spring" autoStartup="true">

                  <dataFormats>
                           <json id="json" library="Jackson" unmarshalTypeName="events.Tweet"/>
                  </dataFormats>

                  <route>
                           <from uri="activemq:topic:twitterEvents" />
                           <unmarshal ref="json"/>
                           <to uri="bean:receiverProcessor" />
                  </route>

                  <route>
                           <from uri="activemq:topic:twitterEventsOld" />
                           <unmarshal ref="json"/>
                           <to uri="bean:receiverProcessor" />
                  </route>
         </camelContext>
</beans>
```

Горната конфигурация извършва следните операции:

1. Инстанциира класа "org.openstreetmap.gui.jmapviewer.JMapViewer " и му назначава псевдоним "map".
2. Инстанциира класа "events.gui.TableModel" и му назначава псевдоним "tableModel".
3. Използвайки обектите от точки 1 и 2, инстанциира обект от класа "events.gui.TabbedFrame" и му назначава псевдоним "tabbedFrame".
4. Използвайки обектите от точки 1 и 2, инстанциира обект от класа "events.gui.ReceiverProcessor " и му назначава псевдоним "receiverProcessor ".
5. Конфигурира JMS компонент, наречен "activemq", насочен към локалната машина и порт 61616.
6. Дефинира формат на данни, който ще използва библиотеката "Jackson" за конвертиране от JSON формат до посочения клас – "events.Tweet".
7. Създава два маршрута, които получават съобщения от JMS темите "twitterEvents" и "twitterEventsOld", конвертират получените JSON обекти

в обекти от класа "events.Tweet" и ги предават на класа "receiverProcessor".

Модула за визуализация на събитията се използва от потребителите на системата, за да наблюдават потока от събития в реално време или повторен анализ на минали събития.

Уеб сървър за визуализация на събития

Уеб сървъра за визуализация на събития представлява инстанция на Jetty уеб сървъра, конфигурирана чрез Apache Camel, която предоставя възможност на крайния потребител да преглежда събитията преминаващи през системата с помощта на уеб браузър.

Модула прави инстанция на Apache Camel със следната конфигурация:

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.springframework.org/schema/beans
         http://www.springframework.org/schema/beans/spring-beans.xsd
         http://camel.apache.org/schema/spring http://camel.apache.org/schema/spring/camel-spring.xsd">

         <bean id="activemq" class="org.apache.activemq.camel.component.ActiveMQComponent">
                  <property name="brokerURL" value="tcp://localhost:61616"/>
         </bean>

         <camelContext id="defaultContext" xmlns="http://camel.apache.org/schema/spring" autoStartup="true">

                  <route >
                           <from uri="activemq:topic:twitterEvents"/>
                           <to  uri="websocket:cameltwitter?sendToAll=true&amp;port=8080&amp;
                                    staticResources=classpath:web/."/>
                  </route>

                  <route >
                           <from uri="activemq:topic:twitterEventsOld"/>
                           <to uri="websocket:cameltwitter?sendToAll=true&amp;port=8080&amp;
                                    staticResources=classpath:web/."/>
                  </route>
         </camelContext>

</beans>
```

Горната конфигурация извършва следните операции:

1. Конфигурира JMS компонент, наречен "activemq", насочен към локалната машина и порт 61616.
2. Създава два маршрута, който получават съобщения от JMS темите "twitterEvents" и "twitterEventsOld" и ги препращат към един и същ WebSocket с псевдоним – "cameltwitter", намиращ се на порт 8080 и съдържащ ресурсите от директорията "web".

Тази конфигурация кара Apache Camel да създаде инстанция на Jetty уеб сървъра очакваща заявки на порт 8080. Когато сървъра получи заявка на този порт, той предоставя ресурсите от директорията web на модула. В тази директория е разположен един HTML файл, който съдържа структурата на уеб страницата представена на клиента и JavaScript код, който извършва следните функции:

1. При отваряне на страницата от браузъра на клиента, се създава Google Maps карта и се разполага в горната част на страницата.
2. Изчертава се празна таблица, в която по-късно ще бъдат попълнени данните от получените съобщения.
3. Отваря се връзка към WebSocket псевдонима "cameltwitter", през който потребителя ще получава постъпващите събития.

При получаване на събитие от страницата, JavaScript кода конвертира JSON обекта в нормален JavaScript обект и добавя данните от събитието във таблицата. Ако събитието съдържа геолокационни данни, то върху картата се добавя и маркер, с необходимия цвят в зависимост от оценката на текста.

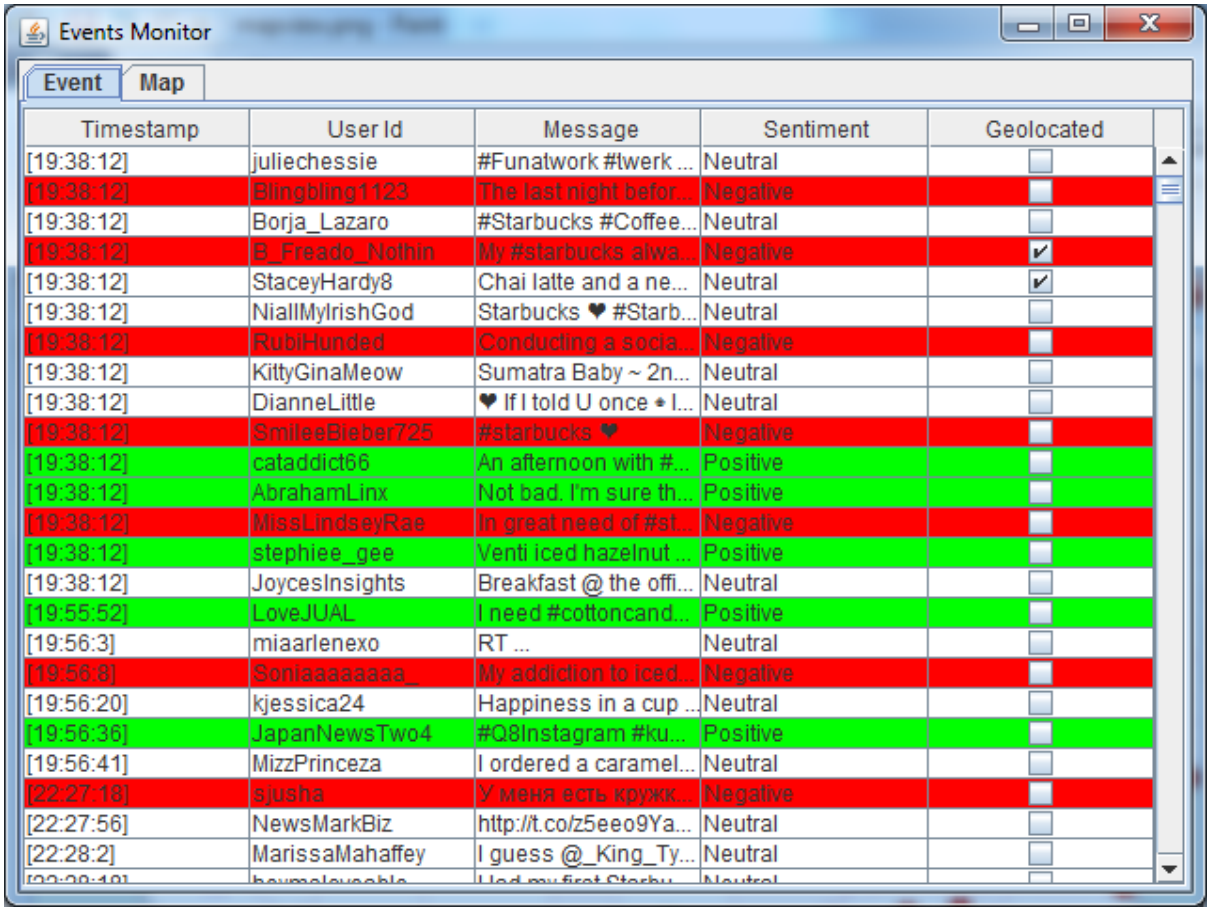
Описание на нормалния процес на работа на системата

При нормалния процес на работа на системата работят всички модули с изключение на модула за възпроизвеждане на съобщения. Нормалния процес на работа включва следните стъпки:

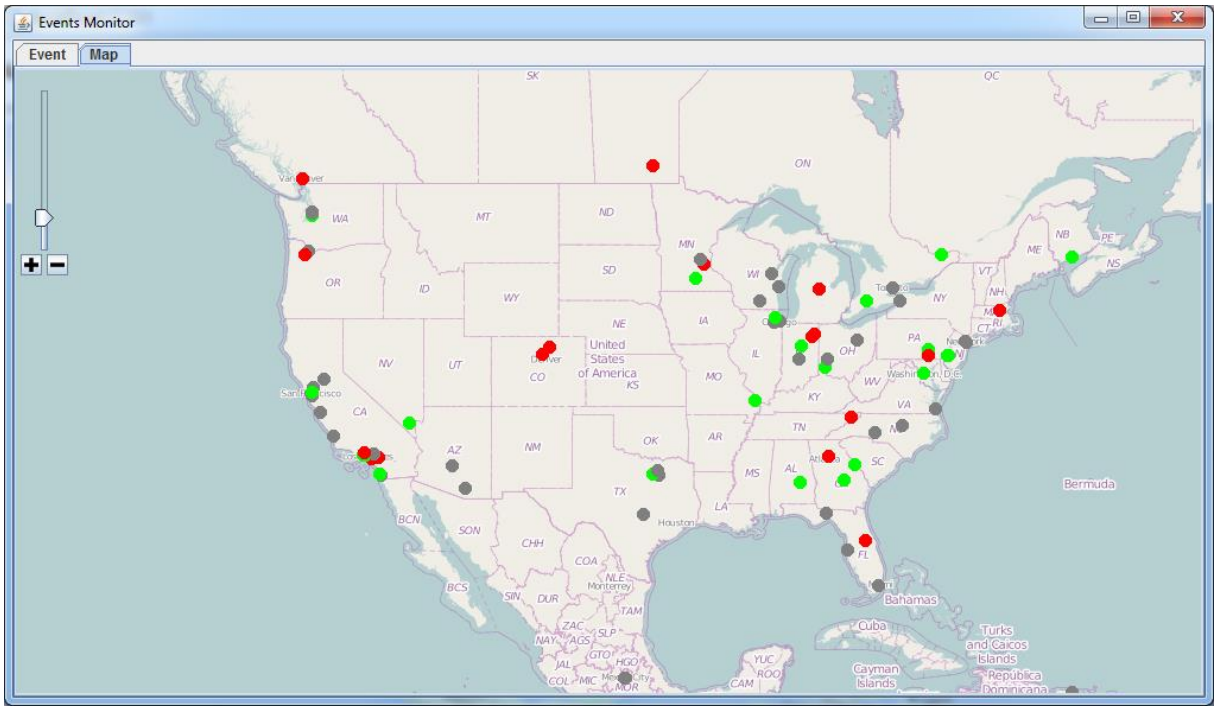
1. Потребител на социалната мрежа Twitter публикува съобщение с текст, насочен към някой от продуктите или самата марка Starbucks (текста на съобщението съдържа "#starbucks").
2. Модула за връзка със външни системи и анализ получава ново събитие, което съдържа данните на изпратеното в точка 1 съобщение. Събитието бива обработено, като от него се извличат необходимите данни, анализира се текста на съобщението, като му се поставя оценка "положителен", "неутрален" или "отрицателен". Преди текста да бъде анализиран от него се премахват специалните символи и URL препратките – тази стъпка е нужна, тъй като потребителите на социални мрежи много често използват емотикони или изпращат снимки а тези символи са невалидни за текстовия

анализатор и пречат на правилния анализ. След като приключи обработката на събитието, данните се конвертират в JSON формат и се препращат към канал за пренос на събития.

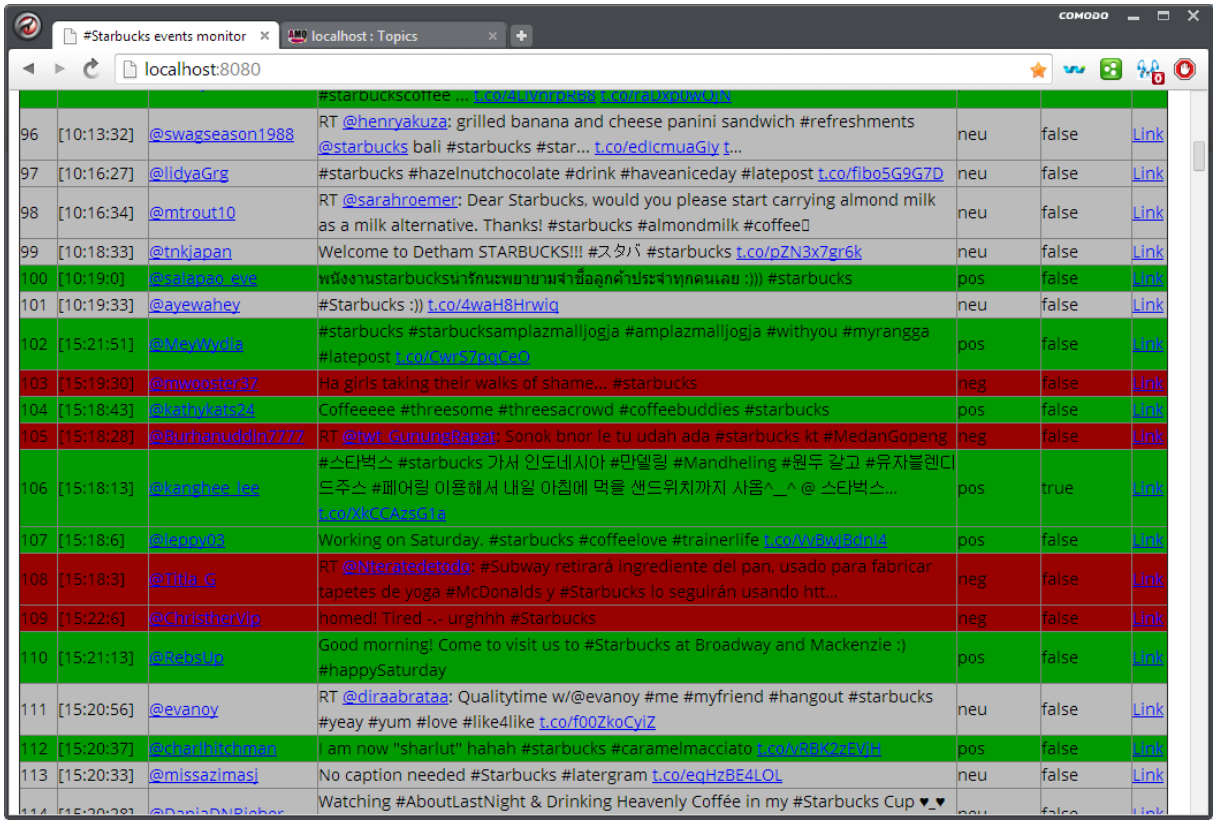
3. Канала за пренос на събития доставя събитието до всички активни абонати.
4. Модула за запис на събитията получава събитието, в резултат на което нов файл, съдържащ информацията от събитието се записва на локалната файлова система.
5. Модула за визуализация получава събитието, конвертира го от JSON формат в стандартен Java обект и представя на крайния потребител данните в табличен вид:



Ако събитието съдържа геолокационни данни, то върху географската карта на приложението се поставя точка със зелен (позитивен), сив(неутрален) или червен(негативен) цвят, в зависимост от оценката на текста на съобщението.

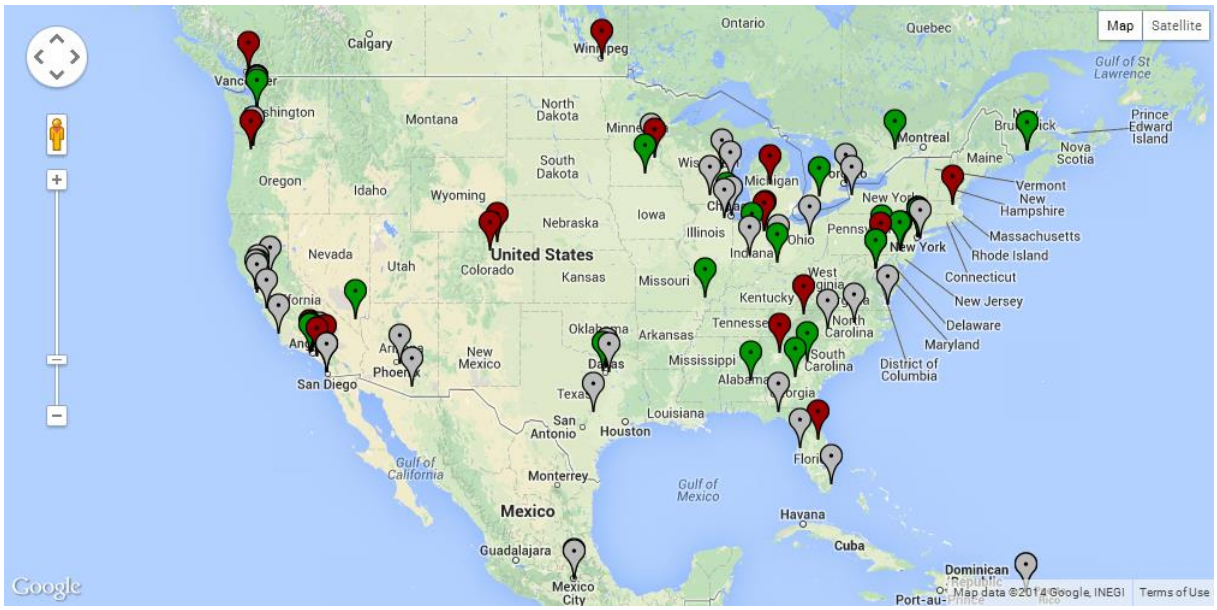


6. Модула съдържащ Уеб сървъра за визуализация получава събитието – полученото от JMS темата събитие се изпраща до WebSocket-а, който от своя страна препраща събитието до всички активни уеб браузъри. JavaScript кода на страницата продължава обработката на съобщението като го конвертира от JSON формат във JavaScript обект и с получените данни попълва нов ред в таблицата на страницата.



			#starbackscoffee ... t.co/4JVmrouGg3 t.co/r6DXqpWQJN			
96	[10:13:32]	@swagseason1988	RT @henryakuza: grilled banana and cheese panini sandwich #refreshments @starbucks bali #starbucks #star... t.co/edicmuaGiy t...	neu	false	Link
97	[10:16:27]	@lidyaGrg	#starbucks #hazelnutchocolate #drink #haveaniceday #latepost t.co/fibo5G9G7D	neu	false	Link
98	[10:16:34]	@mtrout10	RT @sarahroemer: Dear Starbucks, would you please start carrying almond milk as a milk alternative. Thanks! #starbucks #almondmilk #coffee	neu	false	Link
99	[10:18:33]	@tnkjapan	Welcome to Detham STARBUCKS!!! #スタバ #starbucks t.co/pZN3x7gr6k	neu	false	Link
100	[10:19:0]	@sejepao_eye	พนังงานstarbucksนารักนะพยายามจำชื่อลูกค้าประจำทุกคนเลย :))) #starbucks	pos	false	Link
101	[10:19:33]	@ayewahey	#Starbucks :)) t.co/4waH8Hrwiq	neu	false	Link
102	[15:21:51]	@MeyWydia	#starbucks #starbucksamplazmalljogja #amplazmalljogja #withyou #myrangga #latepost t.co/CwrS7poCeQ	pos	false	Link
103	[15:19:30]	@mwooster37	Ha girls taking their walks of shame... #starbucks	neg	false	Link
104	[15:18:43]	@kathykai624	Coffeeeee #threesome #threesacrowd #coffeebuddies #starbucks	pos	false	Link
105	[15:18:28]	@Burhanuddin7777	RT @twi_GunungRapat: Sonok bnor le tu udah ada #starbucks kt #MedanGopeng	neg	false	Link
106	[15:18:13]	@kanghee_lee	#스타벅스 #starbucks 가서 인도네시아 #만델링 #Mandheling #원두 갈고 #유자블렌디드주스 #페어링 이용해서 내일 아침에 먹을 샌드위치까지 사옴^_^ @ 스타벅스... t.co/XkCCAzsG1a	pos	true	Link
107	[15:18:6]	@leppy03	Working on Saturday. #starbucks #coffeelove #trainerlife t.co/WyEwJBdni4	pos	false	Link
108	[15:18:3]	@Titia_G	RT @Niteratedetodo: #Subway retirará ingrediente del pan, usado para fabricar tapetes de yoga #McDonalds y #Starbucks lo seguirán usando htt...	neg	false	Link
109	[15:22:6]	@ChristherVip	homed! Tired -.- urghhh #Starbucks	neg	false	Link
110	[15:21:13]	@RebsUp	Good morning! Come to visit us to #Starbucks at Broadway and Mackenzie :) #happySaturday	pos	false	Link
111	[15:20:56]	@evanoy	RT @diraabrataa: Qualitytime w/@evanoy #me #myfriend #hangout #starbucks #yeay #yum #love #like4like t.co/f00ZkoCyiZ	neu	false	Link
112	[15:20:37]	@charlnitchman	I am now "sharlut" hahah #starbucks #caramelmacciato t.co/vBBK2zEVjH	pos	false	Link
113	[15:20:33]	@missazimasj	No caption needed #Starbucks #latergram t.co/eqHzBE4LOL	neu	false	Link
114	[15:20:28]	@DaniaDNBieber	Watching #AboutLastNight & Drinking Heavenly Coffée in my #Starbucks Cup ♥_♥	neu	false	Link

Ако съобщението съдържа геолокационни данни, JavaScript кода, използвайки приложно програмния интерфейс на Google Maps поставя маркер върху картата в горната част на страницата.



Ако се наложи системата да бъде спряна или потребителя желае да прегледа някое от старите събития, той може да използва модула за възпроизвеждане на събития,

който изпраща вече записани събития отново към канала за пренос, след което те ще бъдат обработени от модула за визуализация и уеб сървъра за визуализация.

Заключение – анализ на резултатите

В настоящата дипломна работа е представена разработката на система за автоматичен анализ на потребителското мнение от социалните мрежи. Системата успешно изпълнява поставените в заданието цели, като успява да осъществи връзка със социалната мрежа Twitter и да се абонира за възникващи събития в мрежата. След като системата получи събития възникнали в социалната мрежа тя успява да ги анализира и да предостави на своите потребители информацията в удобен за преглед вид.

Приложения в системата алгоритъм за "почистване" на получените текстови съобщения повишава значително успяваемостта на автоматичния анализ на съобщенията – когато библиотеката lingpipe обработва "суровите" текстови съобщения тя успява да категоризира едва около 40 % от тях. Когато входните текстови съобщения бъдат обработени и от тях бъдат премахнати URL препратките, специалните символи и емотиконите, библиотеката lingpipe успява да категоризира над 60 % от съобщенията.

Друга положителна черта на системата е нейната модуларност и възможност за отдалечен достъп – крайния потребител може да наблюдава резултатите от работата на система с помощта на браузър или клиентското swing приложение, докато модулите за обработка и пренос на информацията могат да работят на отделни машини.

Бъдещи подобрения:

Към системата могат да бъдат приложени следните подобрения:

- Добавяне на модули за връзка с други социални мрежи – текущата версия на системата позволява връзка само със социалната мрежа Twitter. Би било добре системата да позволява връзка и с другите големи социални мрежи – Facebook, LinkedIn, Google+ и YouTube. За тази цел може да бъде използван Apache Camel компонента "Camel Social", който в момента е в прототипен стадии.
- Добавяне на модул за конфигурация на системата – в конфигурацията на системата се пази в отделен XML конфигурационен файл за всеки модул. По този начин промяна на положението на модула на пренос на съобщенията например изисква промяна на всички останали конфигурационни файлове.

- Разширяване на способностите за анализ на системата – Spam защита – игнориране на множество съобщения с еднакъв текст и/или изпратени от един и същ потребител в кратък интервал от време; оценка "тежестта" на текстовото съобщение, базирана на броя последователи на автора; възможност за блокиране на автори и др.

Използвана литература:

[1] Michelson B. "Event-Driven Architecture Overview", 2011

[2] Stelzner M., "2013 Social Media Marketing Industry Report", 2013

[3] Gundecha P., Liu H., Mining Social Media: A Brief Introduction, 2012

[4] Bing L., "Sentiment Analysis and Opinion Mining", 2012

[5] Zhang L., "Sentiment Analysis on Twitter with Stock Price and Significant Keyword Correlation", 2013

[6] Importance of Social Media for Small Business, http://maramel.com/blog/?p=529, последно достъпен на 04.06.2014

[7] YouTube Statistics, http://www.youtube.com/yt/press/statistics.html, последно достъпен на 04.06.2014

[8] Forbes, http://www.forbes.com/sites/moneywisewomen/2012/08/08/the-developing-role-of-social-media-in-the-modern-business-world/, последно достъпен на 04.06.2014

[9] Modenesis T., "Leverage Enterprise Integration Patterns with Apache Camel and Twitter", http://java.dzone.com/articles/leverage-enterprise, последно достъпен на 04.06.2014

[10] Anstey J., "Open Source Integration with Apache Camel and How Fuse IDE Can Help", http://java.dzone.com/articles/open-source-integration-apache, последно достъпен на 04.06.2014

[11] Ibsen C., "Camel in Action", 2010

[12] http://en.wikipedia.org/wiki/Google_maps, последно достъпен на 04.06.2014

[13] http://en.wikipedia.org/wiki/Java_Message_Service, последно достъпен на 04.06.2014

[14] http://en.wikipedia.org/wiki/JSON, последно достъпен на 04.06.2014