STUDIA INFORMATICA 2014
Volume 35 Number 4 (118)

Wojciech KIJAS
Silesian University of Technology, Institute of Informatics

FACEBOOK CRAWLER AS SOFTWARE AGENT FOR BUSINESS INTELLIGENCE SYSTEM

Summary. The article describes attempts made to use data collected from online social network in the enterprise data warehouse. During our research we designed and developed sample independent system which can work in Data Staging Area of Data Warehouse to complement customer's data with data from Facebook system using FOAF ontology.

Keywords: Business Intelligence, Online Social Network, Facebook, software agent, dimensional modeling, FOAF ontology

FACEBOOK CRAWLER JAKO SOFTWARE AGENT DLA SYSTEMU BUSINESS INTELLIGENCE

Streszczenie. Artykuł opisuje przeprowadzone badania, mające na celu wykorzystanie danych z sieci społecznościowej typu online w korporacyjnej hurtowni danych. W ramach przeprowadzonych prac zaprojektowany i zaimplementowany został niezależny system pracujący w ramach komponentu Data Staging Area hurtowni danych uzupełniający posiadane dane o klientach o dane pobrane z systemu Facebook z wykorzystaniem ontologii FOAF.

Słowa kluczowe: Business Intelligence, sieć społecznościowa typu online, Facebook, software agent, modelowanie wymiarów, ontologia FOAF

1. Introduction

During recent years can be observed that, together with growing popularity of various types of Internet initiatives, a big piece of people every day activity moved to Internet. A few years ago, only information about known people (politicians, actors, musicians) was available on the web. We could read about their relationships, likes and see some private photos made by paparazzi. Now such a data is available about most of people, who have profile on one of the Online Social Networks (OSN) or blogosphere.

Considering the studies on the concept of "homophily" [1], there is a mechanism in which the similarity of people increases the probability of emergence of relations between them. If people tend to associate and bond with similar others, then using information describing relations between customers for corporate analysis, can be very advantageous.

This paper describes the results of some research in the area of supporting Business Intelligence system, with data extracted from Facebook online social network. The idea was to implement software agent, which would automatically complement customer dimension of data warehouse with data, which obtaining would be very difficult in any other way. In most of cases, data for customer dimension is taken from the CRM software used in the company. Those are some basic data for identification and communication with the customer such as name, address, email and phone number. Gathering some other data, for example data describing customer's relations coming from OSNs, may significantly improve utility of the customer dimension.

2. Business Intelligence systems

2.1. Business Intelligence term

The term Business Intelligence was used for the first time in 1958 by IBM researcher Hans Luhn, who defined it as "the ability to apprehend the interrelationships of presented facts in such a way as to guide action towards a desired goal" [2]. Broader definition was introduced in 1989 by analyst Howard Dresner, who proposed Business Intelligence as an umbrella term to describe "concepts and methods to improve business decision making by using fact-based support systems" [3]. However, the term becomes widely used in the late-nineties of twentieth century. One of the greatest achievements in popularizing the term has Ralph Kimball, known researcher, who discovered data warehouse dimensional modeling methodology, which is now most widely used standard in the area of Business Intelligence. Since Ralph Kimball published the first edition of "The Data Warehouse Toolkit" in 1996, data warehouse based Business Intelligence systems have been embraced by organizations of all sizes [4].

2.2. Data Warehouse and dimensional modeling

Every organization deals with some kind of information. The information can be kept in two forms: the operational records or data warehouse. Operational records contain information created as a result of every day activity of the organization, so it can come, for example, from signing new customers, taking orders or issuing invoices. On the other hand, data warehouse (DW) contains information which is created by integrating data from one or more operational records systems or other disparate sources. Ralph Kimball used very good metaphor to compare operational records system and data warehouse. He said that “the users of an operational system turn the wheels of the organization” while “the users of data warehouse, on the other hand, watch the wheels of the organization turn” [4]. From the user perspective, data warehouse is to allow separating, combining and aggregating data in endless combinations to allow users to use it to support decision making. To meet these demands, data warehouse should contain all the functionality to ensure easily accessible, consistent, properly labeled and credible data.

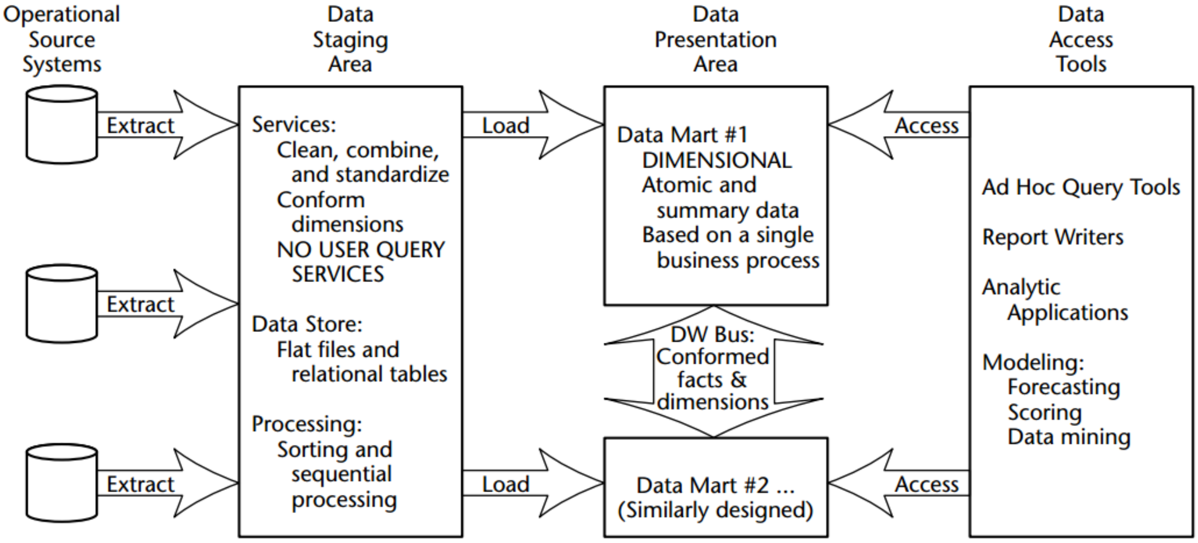

Fig. 1. Typical data warehouse structure [4]
Rys. 1. Struktura typowej hurtowni danych [4]

Typical data warehouse consists of components presented on the figure 1. Operational Source Systems are most often transactional systems of the organization. It is also possible (and this approach is increasingly used) to use external sources of data, which can significantly improve usability of data warehouse. Data Staging Area is an intermediate layer containing everything between operational source systems and the data presentation area. As a metaphor of Data Staging Area we can use car plant, where cars are assembled from raw materials in similar way as raw operational data is transformed to data warehouse. Casual car user can’t enter the car plant before using a car. Otherwise occasional customer would unnecessarily disrupt the work of professionals or be injured with some car plant equipment.

Similarly, data warehouse user doesn't need and even shouldn't be aware of all the processes in data staging area. The layer of data warehouse, which is available to users, is Data Presentation Area, where appropriately structured data is stored and available for querying. All the data access tools used to query, analyze and drill the data, which can be used by data warehouse users for decision making purposes, are referred to as Data Access Tools data warehouse component.

Key assumptions of dimensional modeling are related to Data Presentation Area. The main advantage of dimensional modeling is a simple data model which uses concepts of facts (fact tables) and dimensions (dimension tables) organized in the star or snowflake schema. Fact tables stores most often (but not always) numerical measurements of the reality, while dimension tables stores textual descriptions. Each dimension contains its own primary key, which is used for referential integrity with fact tables, which express many to many relationships between dimensions. Detailed description of the dimensional model vocabulary is available in [4].

Most widely used is Date Dimension as most often business analysis performed using data warehouse are connected with some specific unit of time. For all the customer-centric analysis another critical component of the data warehouse is Customer Dimension. In this paper we will try to show that properly designed Customer Dimension is a big challenge not only because it is most often extremely large, but also because useful data in the Customer Dimension is a cornerstone of utility of data warehouse.

2.3. Supporting Data Warehouse with data from Internet

The idea to use external data from Internet for corporate business analysis is not new. The best-known research in this area was performed by Robert Baumgartner group [7, 8, 9]. Results of their research show some new opportunities, that are coming out together with the growing relevance of the Internet. By developing the Lixto software, they show how the process of collecting and analyzing information about competitors on the market (called "competitive intelligence" [10]) can be nowadays improved by gathering information from new sources such as online web sites or public databases.

In this paper we tried to use online public information not for "competitive intelligence" purposes but, having in mind mentioned studies on the concept of "homophily" [1], we will try to use data available on Facebook online social network to let the organization increase knowledge they have about the customer's relations.

3. Crawling Online Social Networks

Our everyday live is strongly influenced by all kinds of Online Social Networks. For now, total number of monthly active Facebook [4] users is about 1 110 000 000 (June 2013) [6], which means that every seventh citizen of the Earth at least once a month peeks on Facebook. At the same time 680 000 000 [6] are mobile Facebook users. Those statistics shows, that the growing Internet accessibility encourages people to 24/7 online presence. By all their activity, friend's requests, sending messages and sharing likes, they build a solid database of all the relations. It is not surprising that there is growing interest in the OSN data from various backgrounds.

3.1. Crawling Online Social Networks research directions

Due to such a great potential of the data, there is a lot of scientific research connected with OSN. It is possible to list at least two, not necessarily disjoint, directions of the research:

- data collection techniques,
- Online Social Network analysis.

The first category can include the research focused on finding ways to collect data from OSN. We can classify two techniques to collect data from OSN. The first technique relies on usage of API provided by OSN platform itself, while the second relies on HTML parsing. The first technique allows retrieving the data quite easy, but we can rely only on what provides us supplier of the API. In approach presented in [11] authors are using Twitter platform API to collect the data. However not all platforms are as open as Twitter. Using Facebook API we can't access the whole social graph, but only part of the data which is directly connected with our user. For Facebook platform crawling most popular are HTML parsing techniques [12, 13].

In the second category of research connected with OSN we can find all works whose main goal is to discover properties of OSN and apply Social Network Analysis (SNA) to OSN [14, 15, 16].

In current research for crawling Facebook platform data, we decided to implement agent, which walks through Facebook websites and read necessary information by parsing HTML code. So far there was developed a variety algorithms used for producing a representative sample of Facebook users. For this study we used algorithm quite similar but much simpler than Breadth-First-Search (BFS) like algorithms [13]. All the details of our algorithm used to get and maintain Facebook data describing customers are presented in paragraph 4.2.

3.2. FOAF project

The Friend of a Friend (FOAF) project is one of the largest projects on the Semantic Web [17]. The project is creating a Web of machine-readable pages describing people, the links between them and the things they create and do. It has become a widely accepted vocabulary for representing Online Social Networks and many of them use it to produce Semantic Web profiles of their users [18, 19]. In our research we are using FOAF ontology to model and store relations between customers, found by the crawler in Data Staging Area of data warehouse. We decided to use FOAF ontology to take advantage of all the facilities offered by the Semantic Web storage, especially:

- Semantic Web storage is easily extensible and customizable without influence to already stored data,
- Semantic Web storage offers full potential of reasoning over the data that can lead to easy discovery of new connections in data graph,
- FOAF ontology is widely used to describe people and relations so the data can be easily joined with other FOAF knowledgebase,
- Semantic Web storage offers SPARQL [20] query language able to easily retrieve and manipulate data from knowledgebase.

3.3. Software agent

Software agent is computational, goal-oriented system that is capable of autonomous, reactive and proactive behavior. Most often agent reacts to its environment and runs without continuous direct supervision, to perform some function, for an end user or other software [21]. There are a variety of types of software agents (figure 2) depending on type of tasks they perform, environment in which they operate, autonomy and cooperation capabilities.

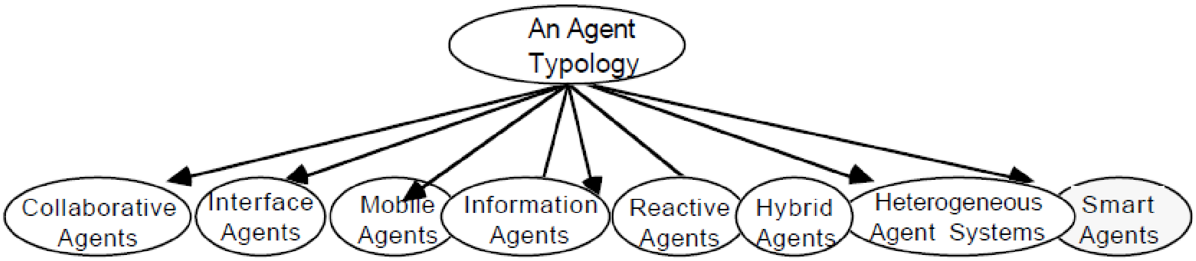

Fig. 2. Classification of Software Agents [20]
Rys. 2. Klasyfikacja agentów programowych [20]

Agent developed during our research can be classified as Information Agent [21]. This category of agents usually exploits Internet and helps manage the vast amount of information in wide area networks like the Internet. We need to endow the Information Agent with the input information where to look, how to find the information and how to collect it, and he just maintains gathering of the information we need.

4. OSN support for building DW Customer dimension

During our research we tried to find a way, to support traditional customer dimension of corporate Data Warehouse, with data describing relations between customers taken from Facebook system. We tried also to find best solution to use the collected data in dimensional model. We focused on collecting only data about friends from Facebook. Collecting data describing other kind of relations, available not only on Facebook but also on other OSNs, will be topic of further research.

4.1. Facebook Crawler components

As part of our research we developed Facebook Crawler system, which responds to the mentioned needs. All the system was developed using Microsoft .NET platform and consists of the components, which are described in details in the following paragraphs. Component model of the system is presented on figure 3.

4.1.1. Facebook Crawler Client component

This component exposes interface for external system. The interface offers two methods which can be used by external system:

- Method used for notification about new customers,
- Method used for querying knowledgebase.

First is the method, which should be called for every new customer added to external system. The method gets new customer e-mail address as parameter and adds new item to the Facebook Crawler Service queue. The second method of the exposed interface is used to query the RDF repository for data gathered by Facebook Crawler Agent. The method accepts query in SPARQL language and use Facebook Crawler Knowledgebase Proxy component to run the query on the knowledgebase.

From the external system perspective (which can be Data Staging Area of Data Warehouse) the Facebook Crawler system is visible only through the prism of those two methods. So external system should notify the agent about any new customer, agent do their work by gathering data from Facebook system to the knowledgebase, external system asks for gathered data using SPARQL query send by second method of input interface, the method returns as much data for the input query as already gathered by agent.

When there are no more new customers to analyze, the agent in the meantime updates data of existing customers, to have all the changes. It is very important process since Online Social Networks data is changing quite often. Detailed description of Facebook Crawler Service algorithm is presented in next paragraph.

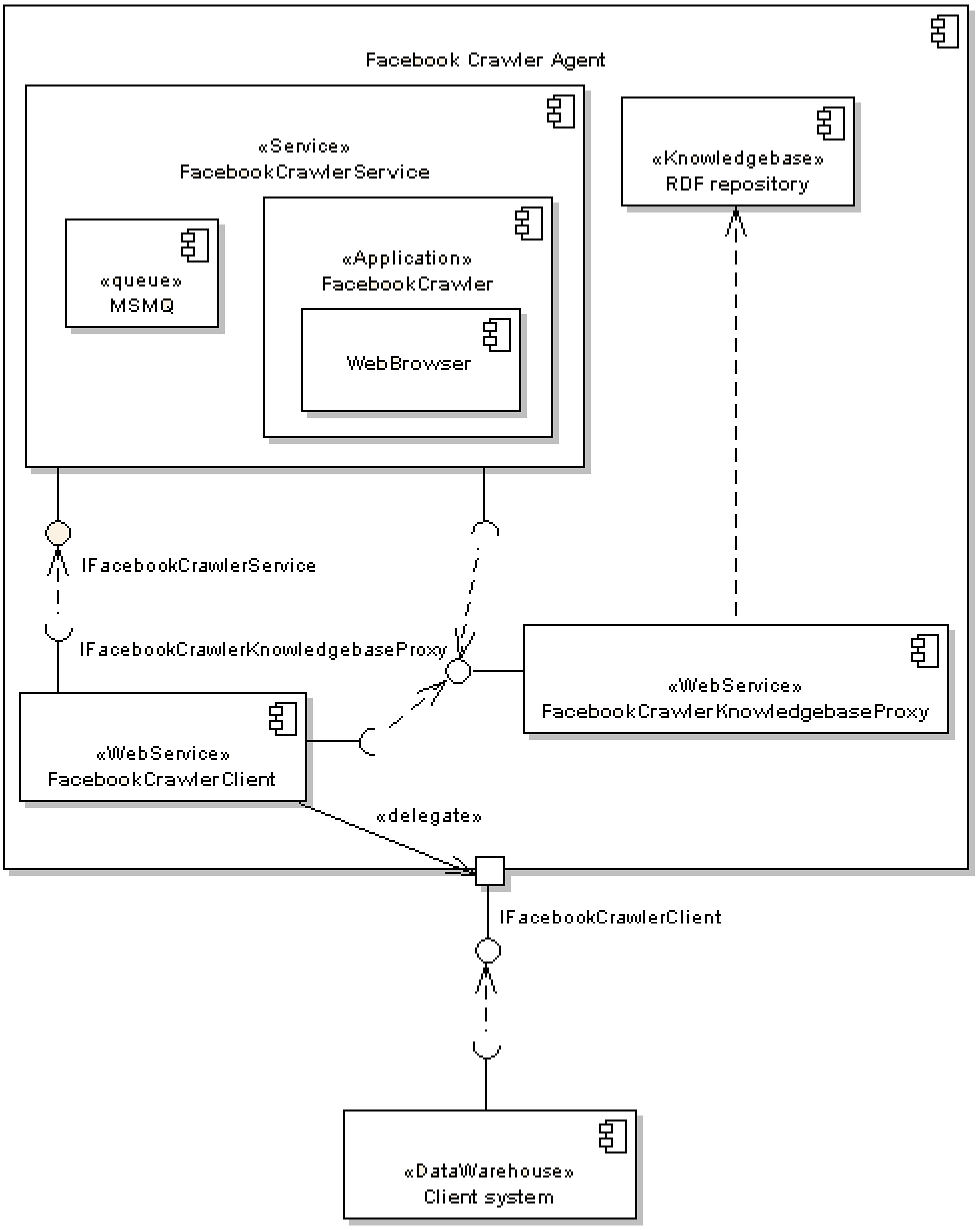

Fig. 3. Components model of Facebook Crawler system
Rys. 3. Model komponentów systemu Facebook Crawler

4.1.2. Facebook Crawler Service component

This component is the heart of the system developed during our research. Input for the service can be FIFO queue developed using Microsoft Message Queuing (MSMQ) [22], or data taken from existing knowledgebase. Depending on data available on the input, it performs different tasks. If there are some requests in FIFO queue, service takes one request and performs new data collection. Otherwise, if there are no any requests in input queue, service takes the oldest entries from knowledgebase and performs data collection for update data with changes. On figure 4 there is algorithm showing the actions performed by service component.

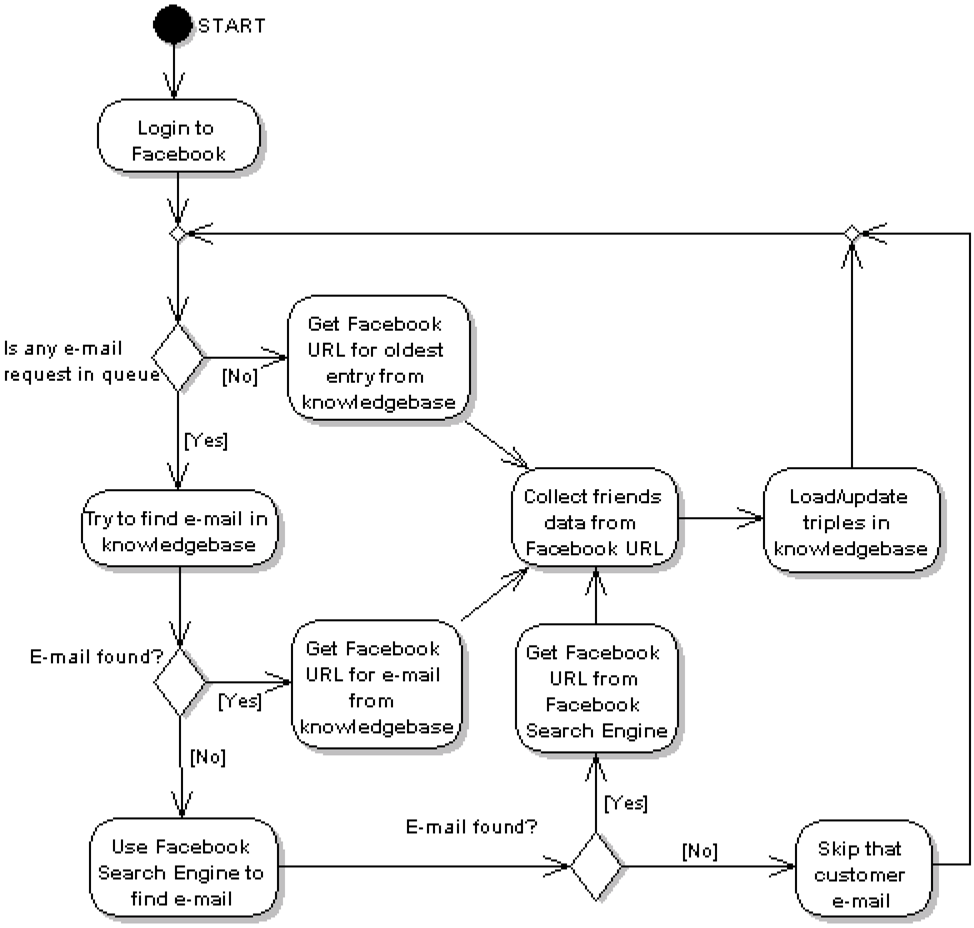
START
Login to Facebook
Is any e-mail request in queue
[No]
Get Facebook URL for oldest entry from knowledgebase
[Yes]
Try to find e-mail in knowledgebase
Collect friends data from Facebook URL
Load/update triples in knowledgebase
E-mail found?
[Yes]
Get Facebook URL for e-mail from knowledgebase
Get Facebook URL from Facebook Search Engine
[No]
[Yes]
E-mail found?
[No]
Use Facebook Search Engine to find e-mail
Skip that customer e-mail

Fig. 4. Facebook Crawler Service algorithm
Rys. 4. Algorytm Facebook Crawler Service

The job of direct crawling of Facebook sites is done by Facebook Crawler application, which is run by Facebook Crawler Service. This is desktop application which takes on input new customer email address, originally coming from external system (for example corporate CRM or directly from corporate Data Warehouse) or existing customer Facebook URL coming from FOAF knowledgebase. Depending on the input data, the application performs appropriate action. All the encountered issues with crawling Facebook sites are described in paragraph 4.2.

The decision to use such a dedicated tool (in our case windows application) to crawl the Facebook website, was made to keep all the Facebook depended logic closed in one component. In that way any major change of Facebook website code results only in the need of upgrading/replacing one small component of the service. From the other hand, with such architecture, we can freely add other similar components with the same interface to extend the system with data from other online social networks.

4.1.3. Facebook Crawler Knowledgebase Proxy component

All the communication with FOAF knowledgebase is performed using Facebook Crawler Knowledgebase Proxy component. The decision to use such a proxy component was made to make the system independent of RDF repository type. Every change of the repository in this case, results only with updating proxy component with new API without any impact on the other system components and interfaces.

4.1.4. RDF repository

The RDF repository is used to store FOAF knowledgebase containing data about relations between customers taken from Facebook system. All the benefits of using FOAF for representing network of users were presented in paragraph 3.2. In current solution we don't use all the potential given by RDF reasoning capabilities. However, we plan to use it in some further research for joining owned repository with other FOAF sources and for semantic integration of data from more than one online social networks.

As part of current research for test of designed system we used two different RDF repositories. One of them was files repository where we keep all the data in raw RDF files. As a second solution we used Dydra [23] graph database working in cloud. Both of the repositories were used only for test so we chose solutions, which don't require additional resources. For the designed system theoretically specific RDF repository type is unrestricted (as long as it provides API for adding and removing RDF statements and run SPARQL queries). However, in specific corporate environment implementation it would be important, to use reliable solution which gives acceptable response times to queries.

4.2. Facebook website crawling

Crawling of Facebook's web front-end was challenging in practice. For logging to Facebook system, HTTP web request with appropriate URL with user name and password was used. To avoid performing login before every request, after first login we serialized given cookies to file and used them for all further requests. After logging to the system, we can easily view profiles of users, who didn't limit the availability of their data using Facebook privacy settings, no matter whether they are our friends or not. However the website use a lot of JavaScript code which makes the page is loaded asynchronously. To get around this problem we decided to use "WebBrowser" .NET Windows Forms control. After programmatically opening user Facebook web page, in the control we programmatically simulate some user actions (scrolling down the page) to load the entire page. After every scroll down, new parts of the page are loaded, we download and parse the actual HTML code of the page and check how many friends are available on the list. On every scroll down additional 20 more friends are presented on the list. Before starting the automatic process there was a need to measure some page load times, to determine time intervals between subsequent scroll downs. For our server (Pentium 4 HT 3.0 GHz, 1.5 GB RAM, Windows 7, and Internet connection 1Mbps) the measured time interval was 16 seconds.

When entire page seems to be fully loaded the process of final parsing HTML code of the page starts. For parsing Facebook web sites HTML code, as a support for LINQ [24] technologies available in .NET framework 4.0, we used external library developed as a part of Html Agility Pack project [25]. For searching interesting data (friend name or friend Facebook URL) in parsed HTML code, we used LINQ queries. The queries were prepared after some analysis of HTML tag structure, so after some Facebook's web front-end update it would be necessary to correct the queries.

The system crawl Facebook sites using simplified (only to the first step) form of the Breadth-First-Search (BFS) [13] algorithm. It is designed in such a way that it collects information about relations only between customers, so there is no need to get and analyze also friends of the friends of current customer. For every new customer system collects data about relations with immediate friends, filters only those who are also customers and creates appropriate FOAF statements in RDF repository. Such a behavior is caused mainly by the fact that current corporation can analyze only data of customers, who gave permission for processing of personal data. Processing data of other casual Facebook users (even if the data is fully available on Facebook site for logged users) could be illegal.

Using test server with configuration mentioned above, the measured time for collecting friends for 1000 Facebook users ranged (depending on average number of friends of the customers) between 20 and 23 hours.

4.3. Using FOAF data in dimensional model

For common Customer Dimension it is enough to use one view for customer's data (figure 5) joined directly with Fact Table. Most often the view contains customer identifier, some identification data such as names, sex, address, email, phone number and other columns, which describe corporate customers and could be interesting from the point of view of the data warehouse user.

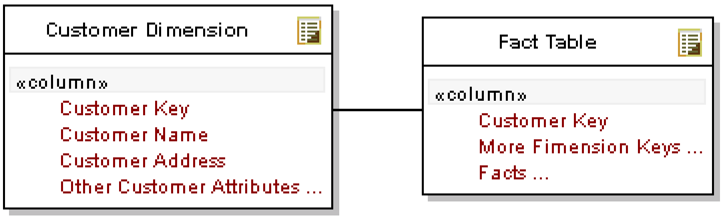

Fig. 5. Common Customer Dimension
Rys. 5. Zwyczajna postać wymiaru klienta

The final stage and the primary purpose of collection of customer data from Facebook, are to use them to extend this corporate dimensional model. There are two main solutions to consider:

- Using the collected data to extend existing customer dimension,
- Using the collected data to create new data mart for customer's relations analysis.

First solution would be implemented by extending one view of customer dimension, with bridge view to store relations between customers. On figure 6 we can see customer dimension and fact table views, the same as in common solution, but between them it is added customer relation bridge view which can be used to join fact table with customer dimension. Use of customer relation bridge is optional, as neither customer dimension view nor fact table view has to be modified in any way, while adding customer relation bridge view. If the bridge table is not used, the customer dimension view joins to fact table in the usual way. Customer relation bridge can be also joined with fact table to create additional view, to make it visible as optional fact table (dotted line on figure 6).

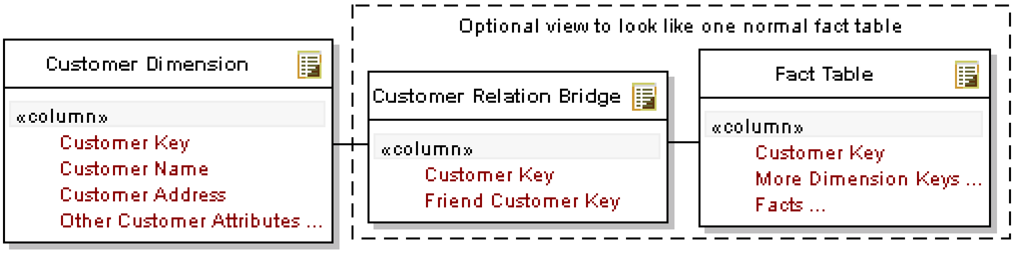

Fig. 6. Customer Dimension with relation bridge
Rys. 6. Wymiar klienta z tabelą pomostową

Using such a solution makes sense in case of some low cost ad-hoc analysis or when we don't have plans to further extend customer relation analysis with some additional dimensions. In that case by adding just one additional bridge table to existing tables and optionally using it to create optional facts view (marked in figure 6 with dotted line) we can make our data warehouse ready for producing some reports using customer relations data.

In the other solution, for customer relations analysis we can use dedicated data mart with sample schema presented in figure 7. Central point of the data mart would be factless fact table, which contain one row for every relation. To have comprehensive information about customer relations, the fact table can be joined with various dimensions. In the presented example we have five dimensions describing customer relations, which is a kind of proposal showing many expansion possibilities of this schema:

- Two dimensions with customer data (Customer Dimension and Friend Customer Dimension) are just views built on one physical table with customer data and reflect two customers which are in relation,
- Date Dimension would reflect date of creation of relationship,
- Relation Source Dimension would reflect source of the information that two of the customers are in relation. In our example so far it would be always Facebook system, but after extending our research with other online social networks it would be more possibilities,
- Relation Type Dimension would reflect type of relation between customers. In our example so far it would be always a common friendship relation, but after extending our research with other online social networks (such as LinkedIn professional network [26]) it would be more possibilities such as professional relations.

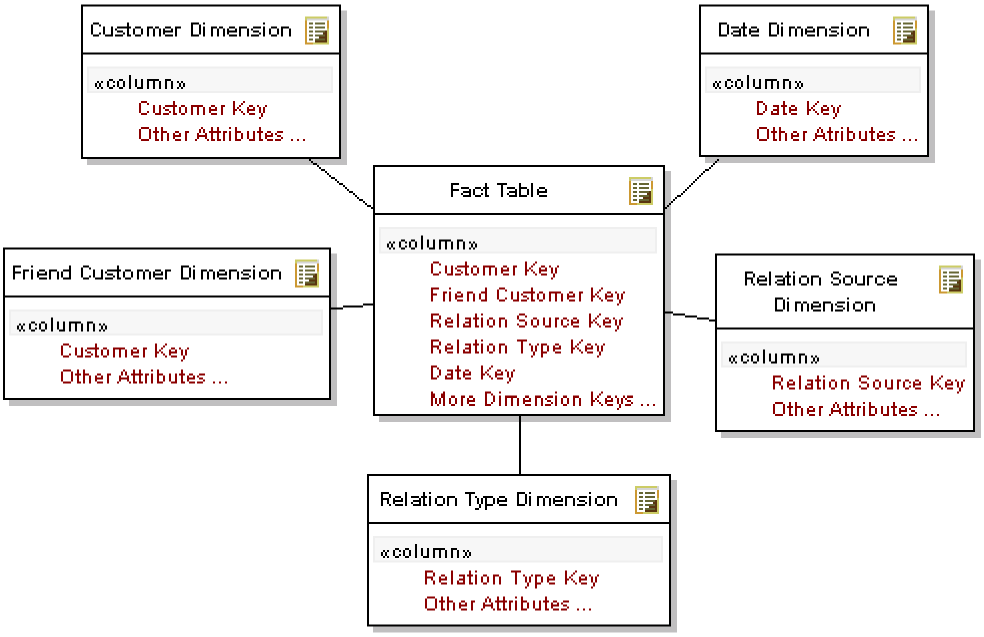

Fig. 7. Data mart for customer relations analysis
Rys. 7. Tematyczna hurtownia danych dla analiz relacji klientów

It must be mentioned that the structure shown in figure 7 contains only variety of data describing customer relations and doesn’t contain any numeric, additive fields that are best thought of as the measurements of the business. For that reason, for some traditional business measurements analysis, it should be joined, using Customer Key, with common corporate dimensional data warehouse, where it would play a role of Customer Dimension.

If the presented Facebook Crawler system is used to carry FOAF knowledgebase in Staging Area of Data Warehouse, nothing stands on barrier to use both of presented solutions simultaneously. For ad-hoc analysis the data can be used to add Customer Relation Bridge in existing data mart while for more advanced analysis new dedicated data mart can be created based on the data. Sample SPARQL query which can be used to get list of connected customers from FOAF knowledgebase to build Customer Relation Bridge can be the following:

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1/>
SELECT ?a ?b
WHERE { ?a foaf:knows ?b }
```

For the second solution (dedicated customer relationship data mart) the mentioned query can be used to get some base data for building factless fact table. Filling the connected dimensions would be performed by separate queries, depending on the specific implementations.

5. Case study

To show practical use of presented system, we employed Facebook Crawler to solve the task of managing data of sample retailer's customers. The retailer used a known marketing trick to gather some basic data of every customer. He announced loyalty program for his customers, where they can collect some points for every shopping and then change some amounts of points for prizes. As a result he was able to gather some basic info for identification and communication with the customer such as name, address, phone number or, what is most important, email. For purposes of the loyalty program the retailer started to use simple CRM system cooperating with cash register system collecting every user basic data and every shopping data. He also encouraged the customers to give him permission for processing of personal data. Having emails of the customers opens a variety of possibilities. Putting the email list as input to Facebook Crawler system allows gathering, from Facebook online social network, knowledge describing connections between customers and make possible doing SNA analysis over graph of connections between customers.

Putting email list as input to Facebook Crawler system, user will get on output FOAF knowledgebase describing connections. To the input of the system user sent 1021 email addresses. The system was working 23 hours and 31 minutes to collect all data for all the customers from Facebook. As not all customers had Facebook account registered for the email, as a result graph of 784 FOAF user profiles was created by the system. As it made a FOAF knowledgebase of around 9000 triples, for better performance, instead of raw RDF files, we used Stardog triple store [27] as repository. We used obtained knowledgebase in two ways:

- to create additional bridge table describing the relations to make possible easily querying the data,
- to perform SNA analysis on the FOAF profiles graph using some additional tools.

5.1. Using bridge table describing relations

Before using Facebook Crawler system to support customer centric analysis, the existing CRM system contained customer dimension table similar to the one presented on figure 5. The table, besides unique Customer Key, contained only basic customer data such as name, address, phone number and email. Some of the more advanced customer centric analysis, which could be performed using the initial database schema, includes only some aggregation of customer transactions using time, product or product category dimensions.

The sample retailer, encouraged to employ Facebook Crawler to his CRM system, didn't want to significantly change the database schema, but just to create some "proof of concept" solution showing the power of customer connections analysis. For that reason we implemented bridge view (similar to the one presented on figure 6) to store relations between customers, which can be used to optionally extend existing customer dimension when it is joined to fact tables. As previously mentioned, use of customer relation bridge is optional, as neither customer dimension view nor fact table view has to be modified in any way while adding customer relation bridge view. If the bridge table is not used, the customer dimension view joins to fact table in the usual way. Customer relation bridge can be also joined with fact table to create additional view, to make it visible as optional fact table (dotted line on figure 6). To fill the relation bridge table we used already mentioned SPARQL query:

```
PREFIX foaf:   <http://xmlns.com/foaf/0.1/>
SELECT ?a ?b
WHERE { ?a foaf:knows ?b }
```

For Stardog triple store running the query took 6 seconds and for the collected 784 user profiles it returned 4468 rows, which corresponds to connections between the profiles. The rows were inserted to Customer Relation Bridge table.

The new Customer Relation Bridge table in the database opens new possibilities of customer centric analysis. The existence of the new table in database can be used to run database queries to answer the following sample questions:

- Get customers who are friends of top 10 customers who bought in recent week the most loaves of new bread which we started to sell a week ago,
- Get 10 recent joined customers together with products which were recent month bought by the highest number of their friends but were not bought by themselves,

- Get common friends of the 10 customers who had the biggest summarized bills over the recent month.

Getting answer to the mentioned sample questions give possibilities to the retailer to take better decisions to whom to address a promotional campaigns to build better relations with the customers with lower costs. To further increase the usefulness of the data and enrich the “proof of concept” solution we decided to use also some basic Social Network Analysis techniques to further analyze collected data.

5.2. Using SNA analysis of FOAF profiles

For SNA analysis of the customer’s profiles we decided to use Node XL [28], which is open-source network visualization and analysis software package. Node XL is very user friendly as it is developed as Microsoft Office add-in [29] that makes it easy to explore graphs, group nodes, calculate some metrics and visualize the results. We imported to the Node XL Excel template the collected 4468 “foaf:knows” connections between the 784 user profiles and used the available class library to calculate degree and betweenness centrality of the graph nodes and to find groups of nodes in the graph.

The node degree is simple metric of undirected graphs showing the number of edges connected to the node. We can say that in the graph of our customer’s connections the degree of the node representing sample customer is the number of their friends who are also customers. The degree can be interpreted in our graph as immediate chance for the customer node of catching whatever is flowing through the network, such as some information about new available products or promotions. From the other side it can be also interpreted as immediate chance of quickly spread the new information over a lot of intermediate connections.

The betweenness centrality of the node reflects number of times a node acts as a bridge in the shortest paths calculated between all pairs of nodes in the graph. Node XL uses, to find betweenness centrality, the algorithm described in the paper "A Faster Algorithm for Betweenness Centrality" by Ulrik Brandes [30]. In our graph, calculation of betweenness centrality can be used to find key customers who play a role of key conduits of information. In our sample data, customer nodes with the highest values of node degree had mostly also high values of betweenness centrality. Most probably it is due to the fact that in the group of people who are customers of the local shop there isn’t any hierarchy similar to hierarchy of some organized and hierarchical networks such as people working in the same company or organized criminal group. On the figure 8 there is presented visualization of the sample network of analyzed customers. Every point represents one customer. The point size is proportional to the value of calculated betweenness centrality of the customer node.

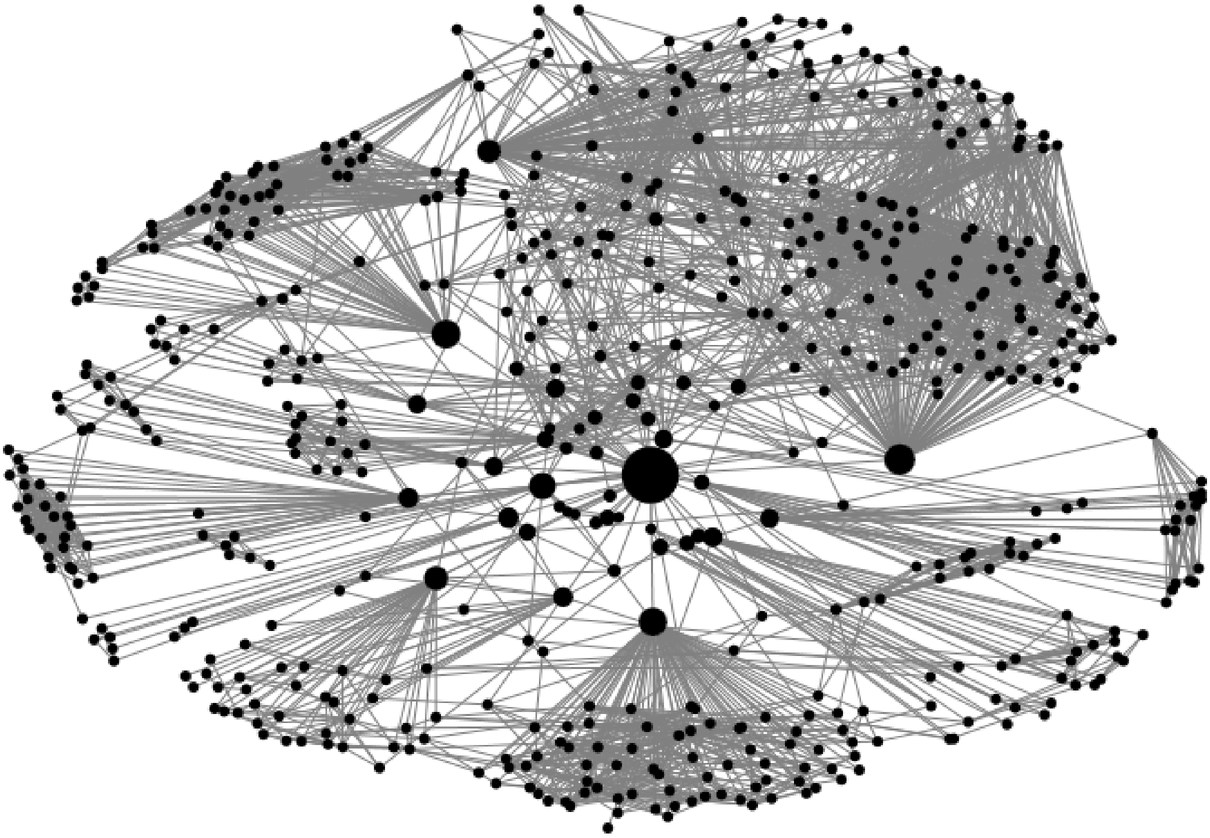

Fig. 8. Graph with marked key customers
Rys. 8. Graf z zaznaczonymi kluczowymi klientami

Nodes in the local retailer's customer graph were also clustered into 14 groups such as that there is a higher density of edges between nodes within a group then between groups. For clustering the graph it was used, implemented in Node XL package, algorithm of A. Clauset, M. E. J. Newman and C. Moor [31]. Knowledge about groups of customers can be helpful for the retailer to better address his promotional campaigns having in mind groups of customers with higher density of edges and key customers in the groups. On the figure 8 there is presented view of the graph with marked calculated clusters.

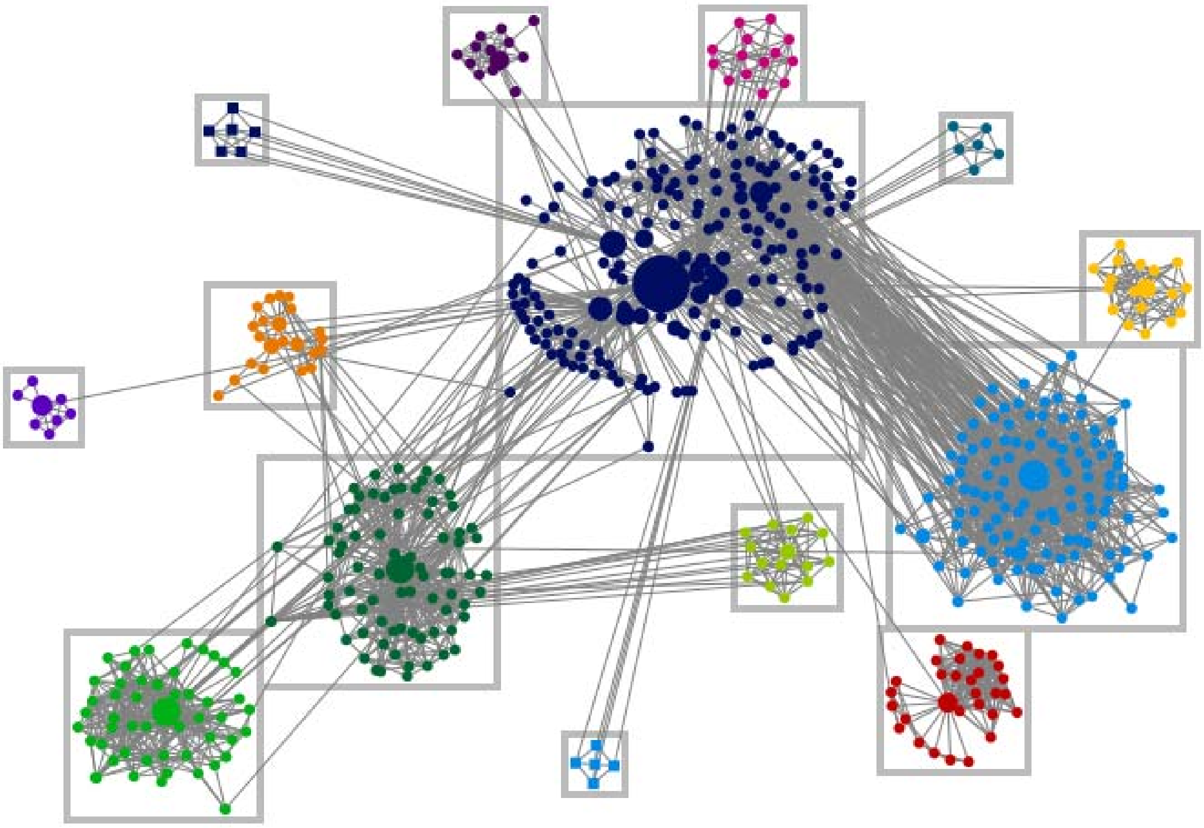

Fig. 9. Graph with marked key customers and clusters
Rys. 9. Graf z zaznaczonymi kluczowymi klientami i grupami

Metrics calculated using external Node XL software could be used to customer centric analysis after inserting them as a three new columns to Customer Dimension table:

- “CustomerDegree” showing calculated degree of the customer node in the graph,
- “CustomerBetweennessCentrality” showing calculated centrality of the customer in the graph,
- “CustomerGroup” showing the cluster to which the customer was added after analysis of density of graph edges.

The existence of such a new metrics for every customer in the dimension table can be used to run database queries to answer the following additional sample questions:

- Get top 20 customers with the highest value of betweenness centrality (who are key conduits of information in the graph of all customers).
- Get top 5 customers from every group of customers, who have the highest value of degree in the group (who have the most friends).

Having answer to the sample questions can be priceless for the retailer. Having in mind the most key customers and most influential customers within different groups, the retailer can take better business decisions even just by listening to their suggestion or needs or can better address promotional campaigns.

6. Conclusion

Online Social Networks appears to be one of the most intriguing phenomenon in recent years. Facebook, with its more than one billion users, plays a prominent role in this field. In this work an attempt was made to take advantage of data collected from Facebook online social network in the enterprise data warehouse. We designed and implemented Facebook Crawler system which plays a role of intermediate layer between Facebook system and Staging Area of Data Warehouse. Facebook data describing relationships among users is not accessible by some public interface, so Facebook Crawler system uses some web extraction techniques to fill RDF repository with FOAF data build based on relations found on Facebook website. In the second step we analyzed some possibilities to use the FOAF data in dimensional model of data warehouse.

As a part of the current work it was also prepared case study showing some practical use of presented system. Using data describing customers of some local retailer, it was prepared sample knowledgebase containing FOAF profiles and connections between them. Data from the sample knowledgebase was used in existing CRM system and was additionally analyzed using SNA techniques to calculate some useful metrics for every customer node to find key customers in the graph and find groups of customer within a graph. The presented solution was a kind of "proof of concept" solution and was performed using relatively small amount of input data. Further development involves creating some tests also on some corporate customer's data to find out how the designed mechanisms prove themselves with the corporate data amounts and how profitable for corporate analysis can be the knowledge about relations between customers. Considering the studies on the concept of "homophily" [1], hawing knowledge about relations between customers, should help to prepare better business analysis and create more productive and better focused advertising campaigns.

For current development we used Facebook as the only source online social network and one server for doing all the job of accepting requests from client application, maintaining FIFO queue, single-threaded crawling Facebook websites and maintaining RDF repository. Planned further development involves extracting data from other popular online social networks, using some multi-threaded solution. Extracting data from networks other than Facebook will allow also using all the potential of RDF reasoning capabilities, to merge instances of the same customer build based on different social networks data.

Current version of Facebook Crawler system allows collecting from Facebook only information about relations between customers. As a part of further development we plan to extend Facebook Crawler knowledge extraction capabilities with extracting same additional user's data such as likes, interests or photos. We also plan to take advantage of Social Network Analysis mechanisms and Data Mining algorithms for analysis of collected knowledge to try to infer, for example, role of the user in community, key users in community or other user's undisclosed attributes using their public attributes or other user's attributes sharing similar interests. Such data can be used to significantly extend and improve the usability of the scheme shown in figure 7.

BIBLIOGRAPHY

1. McPherson M., Smith-Lovin L., Cook J. M.: Birds of a Feather: Homophily in Social Networks. Annual Review of Sociology, 2001, Vol. 27, p. 415÷444.
2. Luhn H. P.: A Business Intelligence System. IBM Journal of Research and Development, 1958, Vol. 2, Issue 4, p. 314÷319.
3. Power D. J.: A Brief History of Decision Support Systems. DSSResources.COM, http://DSSResources.COM/history/dsshistory.html, version 4.0, March 10, 2007.
4. Kimball R., Ross M.: The Data Warehouse Toolkit. Wiley Computer Publishing, 2002.
5. Facebook Online Social Network, http://facebook.com/.
6. Facebook Statistics (verified on 2013-12-10), http://www.statisticbrain.com/facebook-statistics/.
7. Baumgartner R., Frolich O., Gottlob G., Harz P., Herzog M., Lehmann P., Wien T.: Web data extraction for business intelligence: the lixto approach. In Proc. 12th Conference on Datenbanksysteme in Buro, Technik und Wissenschaft, 2005, pages 48÷65.
8. Baumgartner R., Froschl K., Hronsky M., Pottler M., and Walchhofer N.: Semantic online tourism market monitoring. Proc. 17th ENTER eTourism International Conference, Switzerland, 2010.
9. Baumgartner R., Gottlob G., Herzog M.: Scalable web data extraction for online market intelligence. Proc. 35th International Conference on Very Large Databases, 2009, Vol. 2, Issue 2, p. 1512÷1523.
10. Kahaner L.: Competitive Intelligence: How to Gather, Analyze, and Use Information to Move Your Business to the Top. Touchstone Press, 1997.
11. Kwak H., Lee C., Park H., Moon S.: What is Twitter, a social network or a news media? In Proc. 19th International Conference on World Wide Web, Raleigh, North Carolina, USA, 2010. ACM, p. 591÷600.
12. Catanese S., De Meo P., Ferrara E., Fiumara G., Provetti A.: Crawling facebook for social network analysis purposes. In Proc. International Conference on Web Intelligence, Mining and Semantics, Article No. 52, Sogndal, Norway, 2011. ACM.

13. Gjoka M., Kurant M., Butts C.T., Markopoulou A.: Walking in Facebook: a case study of unbiased sampling of OSNs. In Proc. 29th Conference on Information Communications, IEEE Press, 2010, p. 2498÷2506.
14. Mislove A., Marcon M., Gummadi K., Druschel P., Bhattacharjee B.: Measurement and analysis of online social networks. In Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, ACM, 2007, p. 29÷42.
15. Kumar R.: Online Social Networks: Modeling and Mining, The 6th Workshop on Algorithms and Models for the Web Graph (WAW 2009), Barcelona, Spain, http://videolectures.net/waw09_kumar_osnmm/.
16. Kumar R., Novak J., Tomkins A.: Structure and evolution of online social networks. Link Mining: Models, Algorithms, and Applications, Springer, 2010, p. 337÷357.
17. W3C Semantic Web standard, http://www.w3.org/standards/semanticweb/.
18. FOAF Project website, http://www.foaf-project.org/.
19. Golbeck J., Rothstein M.: Linking social networks on the web with FOAF. Proceedings of the 23rd national conference on Artificial intelligence, Vol. 2, AAAI Press, 2008, p. 1138÷1143.
20. W3C Recommendation, SPARQL, http://www.w3.org/TR/rdf-sparql-query/.
21. Hyacinth S. N.: Software Agents: An Overview, Knowledge Engineering Review, Vol. 11, No 3, Cambridge University Press, 1996, p.1÷40.
22. Microsoft Message Queuing technology, http://msdn.microsoft.com/en-us/library/windows/desktop/ms711472.aspx
23. Dydra project website, http://dydra.com/.
24. LINQ language, http://msdn.microsoft.com/en-us/library/vstudio/bb397926.aspx.
25. HTML Agility Pack project website, http://htmlagilitypack.codeplex.com/.
26. LinkedIn Online Social Network, www.linkedin.com.
27. Stardog enterprise graph database website, http://stardog.com/
28. Node XL project website, http://nodexl.codeplex.com/
29. Microsoft Office add-ins development website, http://msdn.microsoft.com/en-us/library/jj620922.aspx
30. Brandes U.: A Faster Algorithm for Betweenness Centrality, http://www.inf.uni-konstanz.de/algo/publications/b-fabc-01.pdf
31. Clauset A., Newman M. E. J., Moor C.: Finding community structure in very large networks, http://www.ece.unm.edu/ifis/papers/community-moore.pdf

Omówienie

W ostatnich latach można zaobserwować przeniesienie dużej części codziennej aktywności wielu ludzi do Internetu. Szczególnie internetowe sieci społecznościowe wydają się być jednym z najbardziej intrygujących i budzących zainteresowanie zjawisk.

W ramach niniejszej pracy zaprojektowany i zaimplementowany został system Facebook Crawler, który spełnia role warstwy pośredniczącej pomiędzy portalem społecznościowym Facebook a warstwą Staging Area korporacyjnej Hurtowni Danych. Zaimplementowany system używa mechanizmów ekstrakcji danych z witryny sieci Facebook do pobrania danych o powiązaniach między klientami i zapisania ich w formie wyrażeń ontologii FOAF do bazy wiedzy zapisanej w postaci trójek RDF. Model komponentów systemu Facebook Crawler znajduje się na rysunku 3, zaś algorytm tworzenia i uaktualniania bazy wiedzy (realizowany przez komponent Facebook Crawler Service) przedstawiono na rysunku 4.

Druga część wykonanych prac polegała na znalezieniu sposobu na wykorzystanie danych FOAF w wielowymiarowym modelu hurtowni danych. Zaproponowano dwa rozwiązania. Pierwsze z nich (rys. 6) polega na dodaniu pomiędzy tabelę faktów a klasyczną tabelę wymiaru klienta, opcjonalnego widoku Facebook Crawler Service, umożliwiającego alternatywne połączenie tych dwóch tabel w zapytaniach wykorzystujących informacje o relacjach. Drugie rozwiązanie (rys.7) polega na utworzeniu dedykowanego schematu przeznaczonego do analizy powiązań między klientami. Centralnym punktem tego schematu jest bezfaktowa tabela faktów i jest ona otoczona szeregiem wymiarów opisującyh powiązania między klientami. Taki dodatkowy schemat może być wykorzystany w dotychczasowej strukturze wielowymiarowej danego przedsiębiorstwa, gdzie po utworzeniu odpowiednich połączeń może pełnić rolę opcjonalnego wymiaru klienta.

Address

Wojciech KIJAS: Silesian University of Technology, Institute of Informatics
Akademicka 16, 44-100 Gliwice, Poland, wojciech.kijas@gmail.com.