ISSN 1392 – 124X INFORMATION TECHNOLOGY AND CONTROL, 2008, Vol. 37, No. 3

AN APPROACH FOR THE FORMATION OF LEVERAGE COEFFICIENTS-BASED RECOMMENDATIONS IN SOCIAL NETWORK *

Lina Tutkute, Rimantas Butleris, Tomas Skersys

Kaunas University of Technology, Information Systems Department
Studentų St. 50, LT–51368 Kaunas, Lithuania

Abstract. On the basis of the drawbacks of the current recommender systems additional functionality to the social network, namely, recommender system with management possibility is proposed. In order to implement such functionality to the social network, the following elements of a new method are presented in this paper: a metamodel of the recommendation; an algorithm for leverage coefficients-based recommendations formation; an algorithm of leverage coefficients-based recommendations interpretation. The paper also discusses the possibility to extend social network's business logic with business rules features and presents some fragments of the prototype that realizes these features.

Keywords: internet social network, recommendation, recommender system, leverage coefficient, business rules.

1. Introduction

With the development of novel technological solutions and constant growth of information quantities, the interaction and communication activities among people and various organizations become more and more computerized. People and organizations form various virtual communities in order to share their knowledge and experience more efficiently. Such virtual communities are the core of social networks [23].

Social network (SN) is a structure that consists of nodes and ties [8]. Nodes define the members of social network (persons, organizations). Ties among nodes identify connections among members of social network [3]. The size of the network is directly related to the size of the community it covers. The transfer of social network into virtual environment (internet social network) enables users to communicate even more efficiently. The main features of such environment are remote (from any place) asynchronous (anytime) communication (information trade) [10].

There are two types of user functions in SN: administrative functions (management of the information about the user, his interest areas, contacts etc.) and functions of participation in SN activities (information upload, search, review etc.) [9].

Social networks usually store huge amounts of information, and that may negatively influence user social actions and reduce possibility to find useful information quickly. However, social network should be able to present not only the main information for the user, but also the additional information that could be potentially useful for the user according to his profile or the actions he performs. One of the ways to present additional information in SN is recommender system (RS) [21].

Recommendation is a description (formal or informal) that defines what additional information should be presented to the user of the social network. Recommender systems (RSs) can be of the different types and complexity. Despite the positive side of RS they are not perfect. Among the main drawbacks of current RS one could mention the following: there is no possibility to modify structure of recommendation or evaluate the environment parameters of the user; also the level of personalization and relevance of the additional information cannot be changed as well.

On the basis of the drawbacks of the current RSs we suggest additional functionality to the social network, namely, recommender system with management possibility. The process of recommendation formation in this system is defined by an algorithm. The particularity of the recommendations formed with this algorithm is that these recommendations are based on leverage coefficients – such a solution allows one to define the most suitable level of flexibility and

* The work is supported by Lithuanian State Science and Studies Foundation according to High Technology Development Program Project "VeTIS" (Reg.No. B-07042).

personalization. Some of the functionality of the Business Rules Management System (particularly, Blaze Advisor) is utilized at this stage – it is used to define calculation formulas of leverage coefficients in the design phase, and the process of leverage coefficients calculation in the phase of recommendations execution; see Sections 4 and 5 for more details on these issues.

The recommendation is stored in the system as a composition of atomic elements – this feature enables the analysis and modification of recommendations, avoiding flexibility and personalization problems that are a common problem in other RSs; the composition of the recommendation is presented in Section 3.

2. Integration of Recommender Systems in Social Network

The types of RS, their formation and integration possibilities into Social Network are analyzed in the following two subsections.

2.1. Advantages and Disadvantages of Methods for Formation of Recommender Systems

The flows of information are intensive and contain lots of information in social networks. That's why there is a lot of additional information that can be potentially useful for the user of the social network. SN augmented with RS provides new useful possibilities to distribute information among the members of the network.

Among other useful functions social networks contain functionality that allows one to inform users about various kinds of news, new users etc. Social network can find not only new information, but also the information that can be potentially useful for the user according to his interest areas set in his profile [18]. Such potentially useful information may contain the most popular articles, goods or services (e.g. if the user is marked in the profile that he likes cats, RS can suggest him to contact others users who are interested in cats or read articles about cats.). All this functionality of the social networks is provided by the recommender systems.

RS can be analyzed in different views. Depending on a view, different classifications may be applied [6]:

- Static and dynamic;
- Depended on the user: his profile [4] or actions [16];
- Automatic [15] and user initiated.

Automatic dynamic recommender systems have the highest advantages. They can have high level of personalization (selection of information according to the profile of the particular user) or high level of relevance (selection of information according to the parameters, which characterize the activity of the particular user in the network). These recommendations are formed using special formation methods [2, 12]. Automatic dynamic RS can implement three types of these methods: content-based [1], based on the communication among users [7, 11, 24], hybrid methods [5, 14]. Some automatic methods have self-learning feature for making better decisions for selecting additional information [2].

It should be pointed out that not all RSs can offer proper functionality [22]. That is why particular usage of the RS described above depends on the specifics of the social network itself and the complexity of the recommendations that is needed.

A summarized comparison of RSs is given in Table 1.

Table 1. Comparison of types of recommender systems (RS)

Type of recommendation / Criterion	Static	Dynamic				
		Partially dynamic		Fully automatic		
		By the action	By the profile	By the content	By the communication	Mixed
Level of personalization	-	+	+	+ (high)	+ (high)	+ (high)
Possibility to change level of personalization	-	-	+	-	-	-
Level of relevance	+	+ (low)	+ (low)	+ (high)	+ (high)	+(high)
Possibility to change level of relevance	-	-	-	-	-	-
Possibility to change structure of presented additional information	+	+	+	-	-	-
Possibility to change criteria for selection of additional information	+	+	+	-	-	-
Demand for gathering additional data	-	+ (high)	-	-	+ (high)	+

The analysis of RSs [22] identified some drawbacks and problems that do not have solutions yet:

- There is no possibility to modify recommendation, unless the program code itself is changed. This, of course, may become very expensive and difficult to accomplish, because the features of recommendation are already in the code [17].
- RS analyzes the connections between the user and the object that will be offered, but other circumstances like time, living place, age, contacts with other users are not evaluated. This results in

poor level of personalization of the recommendation.

- Newly involved users have no connections and no activities. RS has to evaluate that and to offer the additional information with lower level of relevance. There is no possibility to change this level in existing RSs. This problem concerns not only the new but also the mature users of social networks, because high level of relevance between the user and additional information precludes possibility to present additional information of the lower level of relevance.

2.2. Adding Additional Functionality to Social Network

Additional functionality to social network was proposed after the evaluation of the existing methods of recommendations' formation (Figure 1). The core of our proposal is an automatic leverage coefficients-based recommender system of the social network.

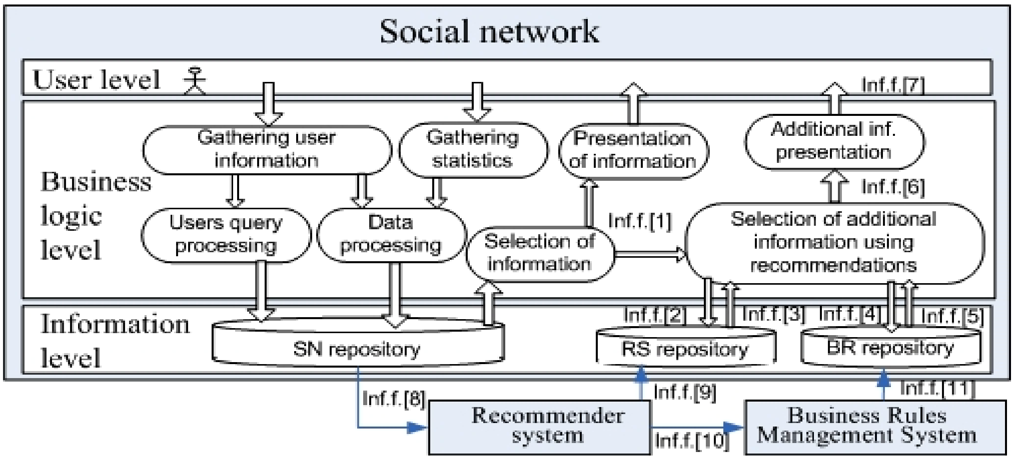

Figure 1. Social network's business logic level augmented with RS and BRMS functionality

The stage of Business logic is augmented with two additional processes: „Selection of additional information using recommendations" and „Presentation of the additional information to user".

The input and output flows of the first process are as follows:

- Inf.f.[1] – initial data, which are selected from the SN repository according to the environmental parameters (user actions, user profile, etc.);
- Inf.f.[2] – queries for the selection of the particular recommendation and its parameters from the repository of the Recommender System (RS). These queries are formed after the analysis of initial data (Inf.f.[1]);
- Inf.f.[3] – data about the particular recommendation and information, which is selected according to initial data. Selected elements of the recommendation are as follows: elements of formula of leverage coefficient, weights, output data structure.
- Inf.f.[4] – queries for the calculation of the results of the particular recommendation from the repository of the Business Rules;
- Inf.f.[5] – calculation results of the data elements according to the formula of leverage coefficient.

Input and output flows of the second process are as follows:

- Inf.f.[6] – additional information that is selected by the recommendation with respect to the values of leverage coefficient;
- Inf.f.[7] – additional information that is customized to the particular user according to his profile settings.

Social network has external connections to the Recommender System and Business Rules Management System (BRMS) [13] [20]. RS forms basic elements of the recommendation (such as initial parameters, output structure etc.). BRMS is used to construct calculation formulas of leverage coefficients. Business rules used by BRMS enable the expert to form and to modify the structure of leverage coefficient more effectively.

Information flows Inf.f.[8]-Inf.f.[11] are used to form inner elements of the recommendation. Inf.f.[8] is a set of data elements from the SN repository, they are used to form the main elements of the recommendation (groups, weights, initial data structure, output structure etc.). Formed elements (Inf.f.[9]) are saved in the RS repository. According to the selected subset of the recommendation elements (Inf.f.[10]), BRMS is used to form the business rules for the leverage coefficient's calculation formula and the calculation itself according to that formula; these business rules (Inf.f.[11]) are stored in BR repository [13] [20]; it can be mentioned that in general such BR repository could be used not only for RS development purposes but in other computerized systems' development activities as well [19]. When the BR project is developed, it is compiled and integrated in the RS.

In order to implement such additional functionality to the social network, the following elements of the new method were proposed and presented in this article: (1) metamodel of the recommendation; (2) an algorithm for leverage coefficients-based recommendations formation; (3) an algorithm of leverage coefficients-based recommendations interpretation.

3. Metamodel of the Proposed Automatic Recommendation

Metamodel of the proposed automatic recommendation is presented and described in this section (Figure 2.).

The elements are as follows:

- One of the elements composing automatic recommendation is the *Informal description* – it specifies the purpose and goals of the recommendation.
- Another element is Initial parameters. Initial data are selected according to those parameters. Initial parameter can be one of the two types: (1) Initial parameter of the system (such parameters define circumstances of when the recommendation should be presented, i.e. user identification, parameters describing user environment); (2) Initial parameter of the recommendation (such parameters define data that should be selected according to the particular recommendation).
- Every selected initial data element gets leverage coefficient value which is calculated by the formula. Leverage coefficient lets the system decide what information according to the appropriate user can be useful to him.
- The formula itself is not stored as an element of recommendation, though elements of the formula are. Formula consists of variables. Variable can be one of the two types: ontology element or query. Values of the variables are digits, i.e. values of ontology elements or results of the queries.
- Formula variables have weights. Weights are defined by the expert. The expert groups ontology elements and gives these groups weights of influence. The expert can also define the structure of the output data, which will be presented to the user. It consists of the elements of ontology. Initial data, output data and leverage coefficients values are received using adequate queries (Selection of initial *data, Selection of output data,* and *Receiving of the variable's value*).

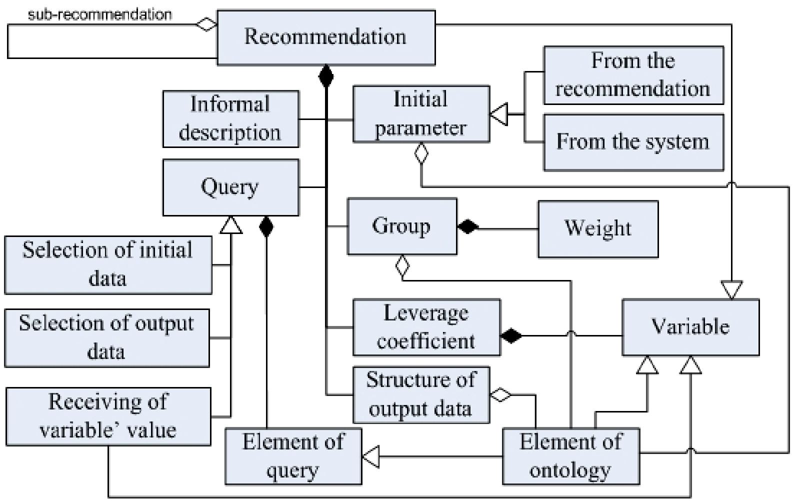

Figure 2. Metamodel of the recommendation

Recommendations are interpreted as a separate element of the SN. What is very important, they are not stored in the program code any more. This feature enables the expert to modify recommendations if needed – it is very useful in social networks (or in other web sites) where dynamics of information flows is very high.

4. The Algorithm of Leverage Coefficients-based Recommendations formation

The core element of the leverage coefficients-based recommendations formation method is the formation algorithm itself.

It is important to note that the same principles for recommendations formation should be applied to all recommendations in the system in order to avoid overlapping recommendations. The amount of recommendations in social network is not finite; it depends on the demand to publish additional information as well as on the specifics of problem domain itself. The usage of system recourses has to be estimated by the expert according to the complexity and amount of recommendations and density of updates. In order to optimize the usage of the system resources, formulas of leverage coefficients may be optimized, eliminating all unnecessary elements. The timing of calculation of leverage coefficients values should also be taken into account – calculations can be performed in real time or periodically.

The essence of the algorithm is the process of calculation of the values to the leverage coefficients' that will be used for the selection of additional information. The particularity of this method is that the user gets information, which is selected according to his profile and other parameters characterizing his environment. This feature partly enables the system to restrict the overflow of the additional information given to the user. It is impossible to avoid excessive additional information completely, because the amount of additional information is a very personalized parameter that differs from user to user. Before the recommendation formation process starts, the expert has to analyze the problem domain and define:

- users or groups of users,
- goals of recommendation,
- composition of recommendation,
- period of data update for recommendation.

After the evaluation of these aspects, recommendation formation process may begin. Basic steps of the algorithm of leverage coefficients-based recommendations formation are presented in Figure 3.

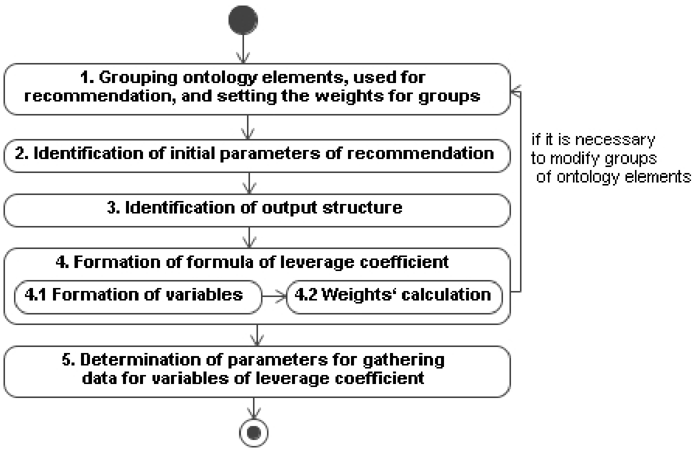

Figure 3. The algorithm of leverage coefficients-based recommendations formation

Let us briefly discuss basic steps of the algorithm:

STEP 1. Grouping ontology elements used for recommendation, and setting weights for groups

The expert analyses problem domain and the description of informal recommendation and groups the elements of ontology of problem domain. This step is made using Recommender System. Some remarks:

- The expert selects the ontology elements that will be used in formation of recommendation. Later, in case there are any changes, expert can easily modify these groups (add new elements, groups, delete them);
- Groups get weights. The weight shows the influence of every grouped element, which means that elements from different groups will have different influence in the formula of leverage coefficient.

The groups are made for every recommendation, i.e. they are not the same for all of recommendations, because even the same elements in different recommendations can have different weights (influence).

In order to avoid an overlap of ontology elements, ensure the correct performance of recommender system and reduce inaccuracy of calculating of leverage coefficients' values some restrictions for forming the groups in the system are imposed:

- The expert cannot put the same ontology element into two or more groups. If he wants to change the weight of the element, he should move it to another group or change the weight of the entire group;
- When the expert deletes element or entire group, he should pay attention if they are already involved in some formula of leverage coefficient, because in case of element deletion the formula may get incorrect;
- The expert can set weight for each group of elements, but the sum of weights in all groups (per every recommendation) should be equal to one. The amount of elements in each group is not limited.

STEP 2. Identification of initial parameters of recommendation

The second step is to identify the initial parameters of the recommendation, i.e. a person who (and when) will get additional information. This step is made using Recommender System. In order to do that, expert has to analyze the following aspects:

- Under what conditions (*when?*) the additional information is presented to the user? It can be some kind of user actions, e.g. opening a particular page of website. According to that the system presents one or another recommendation;
- What parameters identify the adjustment of recommendation to the user? This aspect identifies *who* will be the receiver of additional information.

Some additional parameters, which define user's environment, are identified as well – these parameters enable to personalize the additional information.

The first aspect is analyzed during the integration of recommendation, and the second one – during the formation of the recommendation. These parameters are input data (*initial parameters*) to the recommender system from the main system. There can be general recommendations in the system, which do not have parameters about the user identification, but additional parameters from the main system about the user environment can be given to recommender system anyway.

The expert also controls the selection of initial data (Figure 4) in this step. RS selects data on the basis of the queries formed by the expert. Queries are formed using initial parameters from STEP 2. Selected data define a range of the recommendation ("B" subset in Figure 4).

Remark: Figure 4 represents three sets of data: set A is an entire set of data of the problem domain; B is a subset of A – this subset is formed using initial parameters from the system and from the recommendation (see Step 2); C is a subset of B – this subset is formed using values of leverage coefficients and represents selected additional information that will be presented to the user.

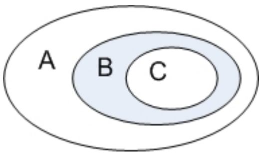

Figure 4. Selection of the Data subset for the recommendation

STEP 3. Identification of output structure

When the initial parameters of recommendation are defined (input parameters), the expert, using the Recommender System, has to define output structure of additional information, which will be presented to the user. Recommendation does not control user interface, nevertheless, it gives a set of data elements to the system, which is responsible for the user interface.

Elements of the output structure can be of the two types: *main* and *additional*. The main elements are related to those ontology elements, which have leverage coefficient's values, because according to those values, additional information is selected. Additional element for the output structure can be any ontology element.

STEP 4. Formation of formula of leverage coefficient

When the first three steps are finished, the expert can form the formula of leverage coefficient using business rules in the business rules management system (BRMS). One defines the elements of the formula and calculates leverage coefficient value for every selected data element using BRMS. Every recommendation has its own formula for calculation of leverage coefficient value. Depending on the complexity of recommendation, there can be one or more elements in the formula. The complexity of the element (which can be: ontology element, query or function) also depends on the requirements for the recommendation.

The structure of the formula for the leverage coefficient is given below:

$$\begin{aligned} f(obj) &= a \cdot f_1(obj) + b \cdot f_2(obj) + \dots \\ &+ v \cdot f_{n-1}(obj) + z \cdot f_n(obj) \end{aligned}, \tag{1}$$

where:

obj – selected element, for which RS has to calculate value of leverage coefficient;

$f_1(obj), \dots, f_n(obj)$ – variables (functions) of the formula;

a,...,z – weights of formula elements; weights show the influence of the particular variable in the formula;

f(obj) – value of the leverage coefficient, which is calculated for the element *obj*.

The expert gets variables from the informal description of recommendation. Weights of formula elements are inherited from the groups of these elements.

Formation of variables. The expert can get information about proper variables in the formula from an informal description of the recommendation. This step is very important, because the proper choice of variables enables the RS to present the additional information to the user with the most appropriate personalization and relevance level. The formula has to be optimized eliminating all unnecessary elements, which can complicate the selection of additional information.

Variable of the formula can be one of the three types:

- Ontology element;
- Query, which returns numerical value;
- Function, which according to the initial parameters returns numerical value.

Ontology element used in the formula must have numerical value. Value can represent:

- Some kind of quantity value (e.g. the number of user visits);
- Object's quality value (e.g. rating of the article). Every object may have different quality values depending on which recommender system selects the most suitable objects as candidates to be presented as the additional information for the user.

Queries are used when values of the variables are not saved in the system. The expert has to form the query and it will be saved in the system.

Functions are the most complicated variables in formulas of leverage coefficients. Functions are assumed as sub-recommendations. They are used when the values of the leverage coefficients cannot be calculated directly. In this case, the value of the variable is equalled to the value of the leverage coefficient of sub-recommendation (Figure 5). When the recommender system initializes the calculation of leverage coefficients' values, it starts from the lowest level of recommendations (S level in Figure 5) and ends up in the highest level (0 level in Figure 5).

0 level $$f_0(obj_o) = a_0 \cdot u_{01}(obj_0) + b_0 \cdot o_{01}(obj_0) + .. + k_0 \cdot f_{11}(obj_0) + .. + n_0 \cdot u_{02}(obj_0)$$

1 level $$f_{11}(obj_1) = a_1 \cdot u_{11}(obj_1) + b_1 \cdot o_{11}(obj_1) + .. + k_1 \cdot f_{21}(obj_1) + .. + n_1 \cdot u_{12}(obj_1)$$

S level $$f_{s1}(obj_s) = a_s \cdot u_{s1}(obj_s) + b_s \cdot o_{s1}(obj_s) + .. + k_s \cdot o_{s2}(obj_s) + .. + n_s \cdot u_{s2}(obj_s)$$

Figure 5. Formula of leverage coefficient with hierarchy of sub-recommendations

where:

- $f_0(obj_O)$ – the main function of leverage coefficient;
- a_0 – the weight of formula‘s variable. Its index shows the level of (sub) recommendation;
- $u_{01}(obj_0)$ – query to get a numerical value of some characteristics of the object *obj* (e.g. rating of the article). The first index of *u* identifies the level and the second one – query‘s number in the formula;
- $o_{01}(obj_0)$ – ontology element. Its value is selected for the object *obj*; the first index identifies the level and the second one – element‘s number in the formula;
- $f_{11}(obj_0)$ – formula's leverage coefficient of sub-recommendation. The first index identifies the level and the second one – its number in the main formula.

It has to be noted that the expert may use only those ontology elements, which belong to the element groups made in Step 1, when he forms a variable of any type in the formula.

Calculation of weights. The next step after the identification of the formula's variables is the calculation of weights of each of these variables. Weight is calculated according to the ontology elements which belong to the particular variable of the formula. Weights of ontology elements are acquired from the groups of ontology elements. There are some rules for calculating the weights of variables:

- If the variable is ontology element, its weight is the same as the weight of ontology element's group;
- If the variable is query, its weight is calculated as the average of all weights of ontology elements composing the query;
- If the variable is function, the calculation of its weight starts from the lowest sub-recommendation level. Formula of the lowest level has only queries and ontology elements, so their weights are calculated using the first and the second rule. The total weight of sub-recommendation is calculated using formula given below:

$$weight(function) = \frac{s_{u[1]} + s_{u[2]} + \ldots + s_{u[k]} + s_{o[1]} + s_{o[2]} + \ldots + s_{o[n]}}{n+k}, \quad (2)$$

where:

- *weight (function)* – weight of the recommendation (or sub-recommendation);
- $s_{o[n]}$ – weight of the variable that is ontology element;
- $s_{u[k]}$ – weight of the variable that is query;
- *n* – the total number ontology elements;
- *k* – the total number of queries.

STEP 5. Determination of parameters for gathering data for variables

When the expert defines all the elements of the formula of leverage coefficients (weights and variables), he has to set the time period that defines how often the data for each variable have to be refreshed. Data can be refreshed in real-time; however, if the time for execution of queries or sub-recommendations is too long compared to the time needed for selection of data about simple ontology elements, a certain time period may be set for these actions.

5. The Algorithm of Leverage Coefficients-based Recommendations Interpretation

The process of recommendations formation is separated from their interpretation. Such an approach enables the expert to modify the elements of recommendation without interruption of the work of social network. During the recommendation formation process elements are saved in two repositories: Business Rules and Recommender System, and in the interpretation mode recommender system selects proper elements and provides personalized additional information to the user.

Basic steps of the interpretation algorithm are presented in Figure 6.

Social network transfers *initial parameters F1* to the RS (the process of the identification of initial parameters was presented in STEP 2). Another set of *initial parameters* (about recommendation) is acquired from the Repository. The full set of initial parameters

F2 is used for the selection of *initial data F3* (the process of the selection of initial data was presented in STEP 2). For every data element, BRMS (Figure 6.) calculates the *value of leverage coefficient F4*. Flow *F4* is used for the selection of additional information with respect to the information output structure of the recommendation (it is described in STEP 3). The result of recommendation interpretation (*F5*) is transferred to the social network and presented to the user.

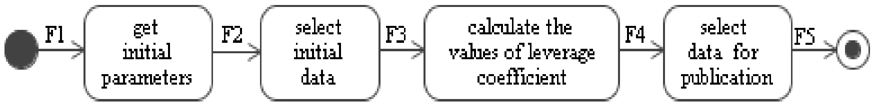

Figure 6. Basic steps of the interpretation algorithm

It should be noted that the interpretation algorithm is the same to all recommendations, which were developed by the proposed method. This feature enables the transportation of the recommendations to other social networks (provided these networks use the same recommendation formation and interpretation algorithms).

6. Prototype of the Recommender System

The proposed algorithm of leverage coefficients-based recommendations formation was implemented in the prototype of recommender management system. This system supports the process of automatic formation of recommendations. Compared to the manual formation of recommendations (when the expert has to write all the program code for the recommendation), the proposed RS reduces time costs of recommendations formation and maintenance, moreover, it reduces the number of human mistakes.

The RS was created using *Jena* framework, programming tool *Eclipse*, data repository *PostgreeSQL*, *Altova Semantic Works* for the formation of ontology and *Blaze Advisor* for formation of business rules. Ontology was used instead of the traditional relational database in this RS. Ontology was chosen because of its features that let the expert define not only the objects of the problem domain but also semantic relations among them.

An example is given in Figures 7 - 9. The ontology is FOAF (Friend of a Friend) and the recommendation is formed for selection the newest most popular articles in the Social Network.

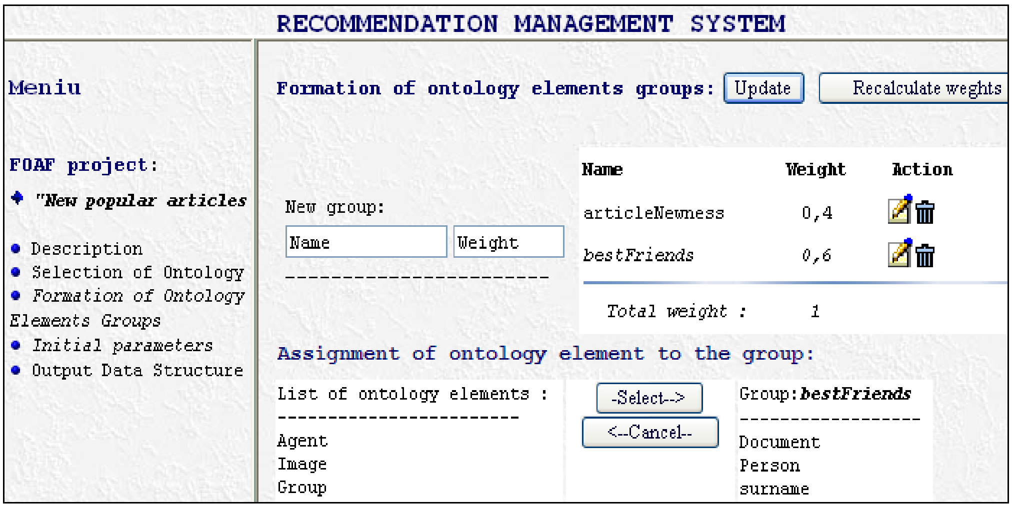

Figure 7. The fragment of the prototype of the RS (Step 1. Formation of ontology elements' groups)

7. Conclusions

On the basis of the drawbacks of the current RS we suggested additional functionality to the social network, namely, recommender system with management possibility. In order to implement such additional functionality to the SN, the following elements of a new method were proposed and presented in this article: (1) metamodel of the recommendation; (2) algorithm for leverage coefficients-based recommendations formation; (3) algorithm of leverage coefficients-based recommendations interpretation.

The particularity of the recommendations formed with the proposed algorithm is that these recommendations are based on leverage coefficients – this feature is realized using business rules and allows one to define the most suitable level of flexibility and personalization. Such recommendations provide proper additional information to the particular user. The recommendation is stored in the system's Repository as a composition of atomic elements – this feature enables the analysis and modification of recommendations, avoiding flexibility and personalization problems that are a very common problem in other RSs.

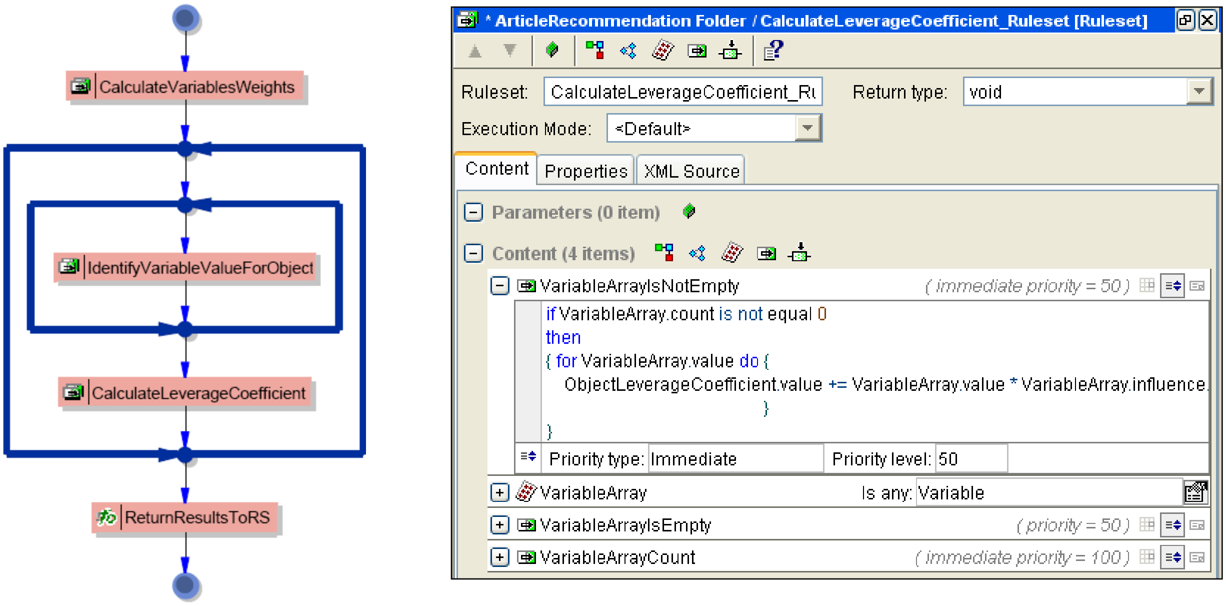

Figure 8. Figure to the left – Algorithm for Calculation of Leverage Coefficients implemented as a RuleFlow in the BRMS Blaze Advisor; Figure to the right – RuleSet that realizes the third Task in the RuleFlow, i.e. CalculateLeverageCoefficient

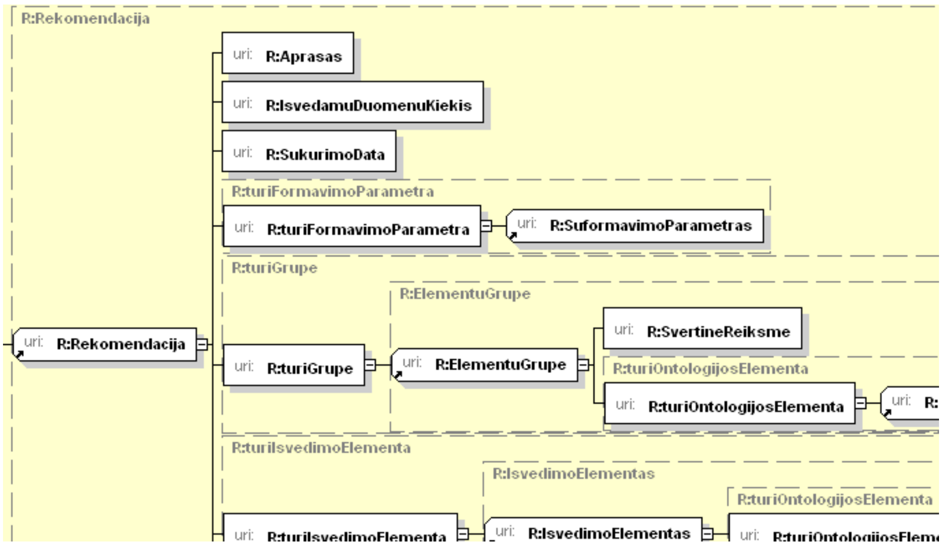

Figure 9. A part of ontology of RS

The modification of recommendations is performed by changing the parameters of elements of recommendation.

The proposed algorithm of leverage coefficients-based recommendations formation was implemented in the prototype of recommender management system. Compared to the manual formation of recommendations (when the expert has to write all the program code for the recommendation), the proposed RS reduces time costs of recommendations formation and maintenance, moreover, it reduces the number of human mistakes. A part of manual work is reduced to the definition of the elements of recommendations assigned to the RS (groups, weights, etc.) and the calculations are assigned to the BRMS (calculation of leverage coefficients values, etc.).

References

[1] **G. Adomavicius, R. Sankaranarayanan, S. Sen, A. Tuzhilin.** Incorporating Contextual Information In Recommender Systems Using A Multidimensional Approach. *ACM Transactions on Information Systems (TOIS), Vol.*23, 2005, 103–145.

[2] **G. Adomavicius, A.Tuzhilin.** Toward the Next Generation of Recommender Systems: A Survey of the State-Of-The-Art and Possible Extensions. *IEEE Transactions on Knowledge and Data Engineering, Vol.*17, 2005, 734–749.

[3] **W. Atteveldt, J.Kleinnijenhuis, D. Oegema, S. Schlobach.** Representing. Social and Cognitive Networks. *Proceedings of the Social Networks Analysis Workshop of the 3rd European Semantic Web Conference* (ESWC'06), 2006,13-26.

[4] **P. Bonhard, M.A. Sasse.** "Knowing me, knowing you" – Using profiles and social networking to improve recommender systems. *BT Technology Journal*, *Vol.*24, Issue 3, 2006, 84–98.

[5] **R. Burke.** Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction*, *Vol.*12(4), 2002, 331–370.

[6] **G. Carenini, J. Smith, D. Poole.** Towards more conversational and collaborative recommender systems. *Proceedings of the* 8*th International Conference on Intelligent User Interfaces, USA*, 2003,12–18.

[7] **K.-W.Cheung, L.F. Tian.** Learning User Similarity and Rating Style for Collaborative Recommendation. *Information Retrieval*, *Vol.*7, *Issue* 3-4, 2004, 395–410.

[8] **J. Cummings, B. Butler, R. Kraut.** The Quality of Online Social Relationships. *Communications of the ACM*, *Vol.*45(7), 2002, 103–108.

[9] **L. Garton, C. Haythornthwaite, B. Wellman.** Studying Online Social Networks. *Journal Of Computer Mediated Communication*, *Vol.*3(1), 1997.

[10] **C. Haythornthwaite.** Building Social Networks Via Computer Networks: Creating And Sustaining Distributed Learning Communities. *K.A. Renninger, W. Shumar (eds.), Building Virtual Communities: Learning snd Change in Cyberspace. Cambridge University Press, UK*, 2002, 159–190.

[11] **J.L.Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl.** Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems* (*TOIS*), *Vol.*22, 2004, 5–53.

[12] **R.Jin, J.Y. Chai, L. Si.** An automatic weighting scheme for collaborative filtering. *Proceedings of the* 27*th ACM SIGIR conference on Research and development in information retrieval, Sheffield, United Kingdom. ACM Press, USA*, 2004, 337–344.

[13] **K. Kapocius R. Butleris.** Repository for Business Rules Based IS requirements. *Informatica*, *Vol.*17, *No.*4, 2006, 503–518.

[14] **G. Lekakos, G.M. Giaglis.** A hybrid approach for improving predictive accuracy of collaborative filtering algorithms. *User Modeling and User-Adapted Interaction*, *Vol.*17, *Issue* 1-2, 2007, 5–40.

[15] **J. Leskovec, A. Singh, J. Kleinberg.** Patterns of Influence in a Recommendation Network. *Proceedings of the* 10*th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD). Singapore*, 2006, 380–389.

[16] **M.R. McLaughlin, J.L. Herlocker.** A collaborative filtering algorithm and evaluation metric that accurately model the user experience. *Proceedings of the* 27*th annual international ACM SIGIR conference on Research and development in information retrieval, Sheffield, UK*, 2004, 329–336.

[17] **E. Pakalnickiene, L. Nemuraite.** Checking of conceptual models with integrity constraints. *Information technology and control*, *Vol.*36, *No.*3, 2007, 285-294.

[18] **Al M.Rashid, I. Albert, D. Cosley, S.K. Lam, S.M. McNee, J.A. Konstan, J. Riedl.** Getting to know you: learning new user preferences in recommender systems *Proceedings of the* 7*th international conference on intelligent user interfaces, San Francisco, California, USA. ACM Press, NY, USA*, 2002, 127–134.

[19] **T. Skersys, S. Gudas.** The enhancement of class model development using business rules. *Advances in Informatics:* 10*th Panhellenic Conference on Informatics, PCI'*2005, *Volos, Greece. LNCS Theoretical Computer Science and General Issues* 0302-9743, *Vol.*3746, *Berlin*, 2005, 480–490.

[20] **T. Skersys, V. Pečiulis, R. Simutis.** Business rules specification using natural language-based templates: approach and implementation. *Information Technologies'*2008: *proceedings of the* 14*th International Conference on Information and Software Technologies* (*IT* 2008)*, Kaunas, Lithuania*, 2008, 353–360.

[21] **L. Stoilova, T. Holloway, B. Markines, A.G. Maguitman, F. Menczer.** GiveALink: Mining A Semantic Network Of Bookmarks For Web Search And Recommendation. *Proceedings of the* 3*rd International Workshop on Link Discovery, Chicago, Illinois, USA*, 2005, 66–73.

[22] **L. Tutkute R. Butleris.** Leverage coefficients-based recommendations formation in social network. *Proceedings of the 6th International Conference on Perspectives in Business Information Research* (*BIR*'2007), *Tampere, Finland,* 2007, 143–156.

[23] **B. Wellman.** Computer Networks As Social Networks. *Science*, *Vol.*293, 2001, 2031–2034.

[24] **B. Xie, P. Han, F. Yang, R.-M. Shen, H.-J. Zeng, Z. Che.** DCFLA: A distributed collaborative-filtering neighbor-locating algorithm. *Information Sciences*, *Vol.*177, *Issue* 6, 2007,1349–1363.

Received June 2008.