



## SOFTWARE FOR DETERMINATION OF BEHAVIOR PATTERNS OF WEB-FORUM MEMBERS

Yuriy Syerov, Ruslan Kravets

National University "Lviv Polytechnics", 12 Bandery Street, Lviv, 79013 Ukraine  
syerov@ridne.net, rkravets@ua.fm

**Abstract:** *article considers actual problem of Web-forums' members' behavior and classifying information system development. In the article existing Web-forum CMS' were analyzed, process of Web-community members' behavior analyzing was researched and overviewed, the process of information system developing was described.*

**Keywords:** *Web-community, forum, internet-forum, Web-community member, members' classes.*

### INTRODUCTION

On the modern day level of Internet and global information system World Wide Web development, especially with appearance of new conception – Web 2.0, very important role is played by Web-communities, which consist of everyday Internet users. Web-forums are one of the most popular and effective, from the point of view of information representation and accessibility, forms of Web-community organization [1, 8].

Primary Web-forum's objects are members and content which they create: threads and posts. Web-forum member is a person that visits Web-forum site, reads, and publishes information in threads and posts. Web-forum content is a hierarchy of threads and posts, tree with two types of nodes: sub-forums and discussions.

Nowadays, most modern Web-forums function on the basis of popular software – Content Management System (CMS), for example: vBulletin, Invision, phpBB, XMB [1-3].

Research and analysis of Web-forums' members' behavior is a very complicated computational task. That is why development of information system of analysis of Web-forums' members' behavior, which should simply be integrated into modern CMS, is a very important task.

Members influence Web-forum's life that is why the Web-community developers need software, which lets them analyze members' behavior that helps to provide effective moderation and administration.

### 1. BACKGROUND

Web-forum functioning is impossible without a program complex, which provides basic tasks such as: member registration, content publishing, content, and member management. Such modern day complexes, such as CMS, have typical data structure scheme and provides all basic means that make it possible for Web-forums to function. The majority of Web-forums are based on popular CMS that have similar interfaces and functionality. This helps Web-forums members to familiarize themselves on new Web-forum quickly. Also, this makes it possible to develop informational system of analysis of Web-forums' members behavior for Web-forum administrators.

Modern CMS have good functionality, but they do not give administrators the means to analyze the Web-forum member's behavior. CMS provides only the simplest statistical information: post count, thread count, quantity of new members that registered during the last day, number of existing members that visited the Web-forum today, etc. The system does not give detailed and more complex information about content growth dynamics, member growth dynamics, or member's usefulness. That is why it is needed to develop a system that will provide the means of analyzing member behavior during certain periods of time, classifying members, provide decision support during Web-forum administration [1-3,8,9].

### 2. ARTICLE AIMS

The purpose of this article is to describe the process of information system of analysis of Web-

forums' members' behavior development.

Before we start system development it is necessary to research modern Web-forum CMS structures and the process of Web-forums' members' behavior analyzing. The reason this is necessary is because, our information system must solve all tasks of members' behavior analysis and be easily integrated into existing Web-forum CMS. On the basis of results of this research, we should plan a system structure and develop its prototype. So, the aims of this article are:

- analysis of the existing Web-forum CMS;
- research of the process of Web-forums member behavior analysis;
- development of data structure scheme of information system;
- program realization of algorithms of the Web-forums' members behavior analysis.

### 3. MAIN PART

#### 3.1. STRUCTURAL AND INFORMATION WEB-FORUM MODEL

Primary Web-forums' objects are members and content, which they create: threads and posts.

Web-forum member is a person that visits Web-forum site, reads, and publishes information in threads and posts.

According to their abilities Web-forums' members belong to one of the following classes: unregistered visitors (guests), registered members, moderators, administrators.

Sub-forum is a set of lower-level sub-forums and threads.

Thread is a set of posts, created by Web-forum members. Threads could be created on any structure level, usually on the lowest level.

Web-forum content is a hierarchy of threads and posts (see Fig. 1).

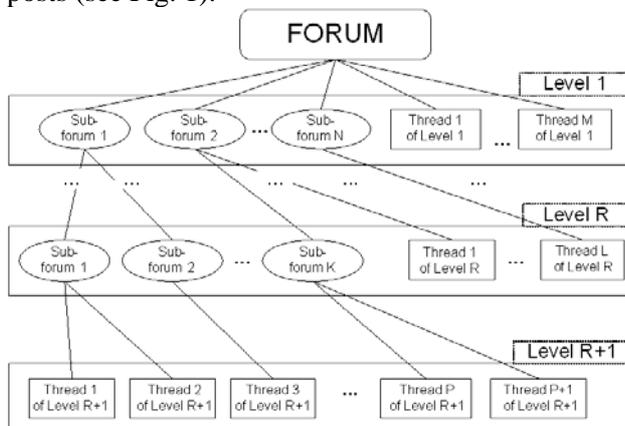


Fig. 1 – Tree-like Web-forum structure

Web-forum consists of a set of first-level forums. Each of them also consists of a set of second-level forums (sub-forums) and threads. Web-forum

structure can be detailed more and more using lower levels. Web-forum structure could be changed during the Web-community's life. If members accumulate great amount of content more sub-forums on the lower levels could be created.

Forum – is a set of sub-forums and threads. Each sub-forum can consist of lower-level sub-forums and threads. In the process of creating forum structure it is recommended to use lower-level sub-forums. Using lower-level sub-forums helps to create better content semantic map.

Thread – is a set of posts, written by forum members. Threads could be placed on any level, but it is better if they are only at the lowest level. Main Web-forum feature is that threads are stored until the forum exists for future members; in some rare specific situations when administrator takes decision to delete one of them.

Poll is a kind of discussion that aims to know member's opinion on some question. Poll start is a question with a set of answers with some explanation. Set of answers must be complete (all available answers). Member can write posts here like in any thread.

Post is a message written by a forum member.

Each post can be one of the following types:

- Thread start – first message in the discussion. It is very important to create correct understandable thread name.
- Answer-post. Each post in the thread that is not thread start. Answer-post may include previous parts of the previous posts – citations.
- Thread stop – last post in the thread, usually made by administrator or moderator when discussion is over or members start to write off-topic or flame posts.

#### 3.2. GENERALIZED WEB-COMMUNITY INFORMATION MODEL

In general all Web-forum CMS have complex data scheme, which consists of great amount of tables. However, large part of these tables is secondary and store data that is not mandatory for Web-forum analysis.

After examining and analyzing existing CMS's, we can select only the main objects of such information system as Web-community. Generalized data scheme, which gives us all necessary information for Web-community member behavior analysis is shown on Fig. 2.

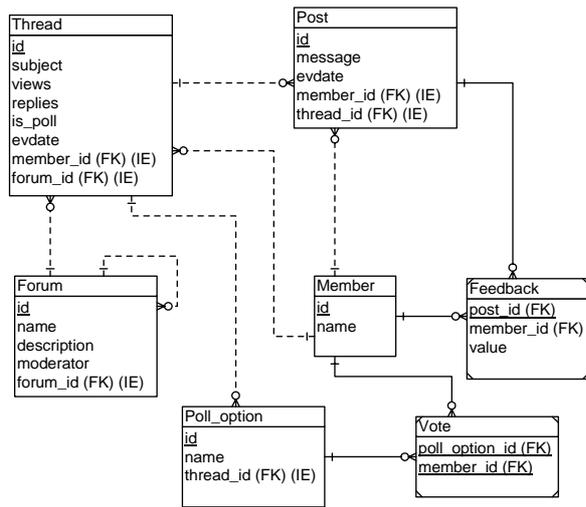


Fig. 2 – Generalized data scheme of web-forum CMS

Main parts of such information system, as Web-forum, are Members and content.

Content consists of Forums, Threads, and Posts.

A Web-forum’s member is characterized by such parameters:

Name, Password, Personal Info, Set of created posts, Set of created threads, Set of received feedbacks, Set of given feedbacks.

Another part of this system – content, is presented by three entities: Forum, Thread, and Post.

Forum is characterized by such parameters:

Name, Number of threads (that belong to this forum), Number of posts (that belong to this forum), Description.

The thread is characterized by the following parameters:

Title, Author, Sub-forum (to which the thread belongs), Number of posts, Number of views.

The post is characterized by the following parameters:

Title, Body, Author, Data, Thread (to which the post belongs), Sub-forum (to which post belongs).

Prior to the system information model description, let us shortly overview Web-forums’ members’ behaviors, their features, and classification methods.

Web-community members may have the following features: Activeness, Creativeness, Attractiveness, Reactivity, and Loyalty.

For Web-community member features modeling, fuzzy sets are used [2]. The features of all listed members can be defined according to their activity within the forum. Activeness is defined by quantity of content that the members have created. Creativeness is defined by the quality of content that the members have created. Attractiveness is defined by the number of members who give feedback to the created content. Reactivity is defined by the way the member takes part in discussions. Loyalty is defined by the reaction to other member’s content.

### 3.3. INFORMATION DATA MODEL FOR CLASSIFYING MEMBERS

The first step of creating information system of web community members behavior analyzing and classifying is development of data structure, mandatory for analysis. In order to create information system, it is necessary to develop metadata structure, which will describe the classes of members, rules of classification, and linguistic variables, which describe the features of members’ behavior.

Developed information model looks as follows (Fig. 3.):

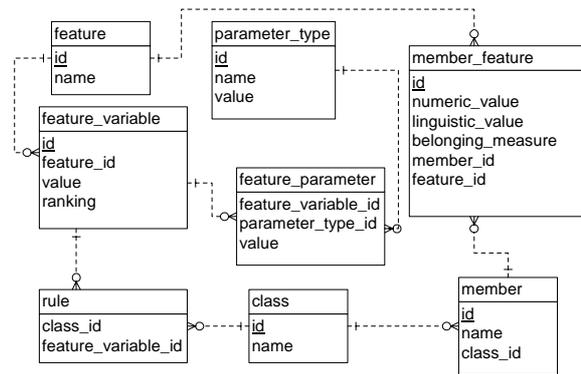


Fig. 3 – Data scheme of an information system of web community members behavior analyzing and classifying

Developed data scheme of information system of Web-forums’ members’ behavior analyzing and classifying will include metadata, mandatory for classification of Web-forum members. It integrates with the database of CMS Web-forum on the basis of entity “Author” and consists of the following tables:

- Table *member* – keeps the information about all Web-community members;
- Table *class* – describes all Web-community members’ classes;
- Table *rule* – describes the rules of Web-forum members’ classification;
- Table *member\_feature* – dedicated to keeping data about all members’ characteristics;
- Table *feature* – describes all Web-community members’ features;
- Table *parameter\_type* – keeps the information about intervals of each linguistic variable.
- Table *feature\_variable* – describes a domain of every linguistic variable – features of a Web-community member.
- Table *feature\_parameter* – keeps the information about range of each linguistic variable.

### 3.4. CALCULATION OF MEMBERS' VALUE CHARACTERISTIC

Members' features will be calculated as follows.

Activeness of threads creation we will present as division of the number of threads, created by the member, by the total number of threads created by all the members:

$$Activeness_{Thread}(member_i) = \frac{card(Thread(member_i))}{card(Thread)}$$

where  $card(X)$  – power of the set  $X$ ;

$Thread(member_i)$  – the set of all threads, created by  $i$ -th member;

$Thread$  – set of all threads.

In the database this computation is realized with the following query:

```
SELECT
  member_id,
  COUNT(*)*100.0/(SELECT COUNT(*)
  FROM Thread) as ThreadActiveness
FROM Thread
GROUP BY member_id
```

Activeness of polls creation is calculated similarly to activeness of threads creation:

$$Activeness_{Poll}(member_i) = \frac{card(Poll(member_i))}{card(Poll)}$$

where  $Poll(member_i)$  – the set of all polls, created by  $i$ -th member;

$Poll$  – set of all polls.

Query for activeness of polls creation looks as follows:

```
SELECT
  member_id,
  COUNT(*)*100.0/(SELECT COUNT(*)
  FROM Thread WHERE is_poll='y') as
  PollActiveness
FROM Thread
WHERE is_poll='y'
GROUP BY member_id
```

Activeness of posts creation we will present as division of the number of posts, created by the member, by the total number of posts created by all the members:

$$Activeness_{Post}(member_i) = \frac{card(Post(member_i))}{card(Post)}$$

where  $Post(member_i)$  – the set of all posts, created by  $i$ -th member;

$Post$  – set of all posts.

Query for activeness of posts creation looks as follows:

```
SELECT
  member_id,
  COUNT(*)*100.0/(SELECT COUNT(*)
  FROM Post) as PostActiveness
FROM Post
GROUP BY member_id
```

Activeness of voting we will present as division of the number of polls, in which the member took part, by the total number of polls created by all the members.

$$Activeness_{Vote}(member_i) = \frac{card(Vote(member_i))}{card(Poll)}$$

$Vote(member_i)$  – all polls in which  $i$ -th member took part;

$Poll$  – set of all polls.

Query for activeness of voting looks as follows:

```
SELECT
  member_id,
  COUNT(*)*100.0/(SELECT COUNT(*)
  FROM Thread WHERE is_poll='y') as
  VoteActiveness
FROM Vote
```

All types of activeness can be calculated during the lifetime or different periods of time, for example by years.

Here is the query for calculating activeness by year:

```
SELECT
  member_id,
  Year(evdate) as Years,
  COUNT(*)*100.0/(SELECT COUNT(*)
  FROM Post) AS PostActivenessYear
FROM Post
GROUP BY member_id, Year(evdate)
```

Attractiveness is calculated in two ways. The first way: attractiveness is the arithmetic mean of all divisions' of members' reactions on the  $i$ -th member's posts divided by the total number of forum's members, divided by the number of  $i$ -th member's posts.

$$Attractiveness(member_i) = \frac{\sum_j^N \frac{card(Feedback(post_j(member_i)))}{card(Member)}}{card(Post(member_i))}$$

where  $Feedback(post_j(member_i))$  – replies of all members to  $j$ -th post created by the  $i$ -th member;

$Member$  – a set of forum members;

$Post(member_i)$  – number of posts created by  $i$ -th member.

Query for the first way of calculation of attractiveness looks as follows:

```
SELECT
  F.member_id,
  COUNT(*)*100.0/(SELECT COUNT(*)
  FROM Member)/(SELECT COUNT(*) FROM
  Post WHERE P.member_id=F.member_id)
  as Attractiveness
FROM Feedback as F
GROUP BY F.member_id
```

The second way of calculating the attractiveness is: as a relation of the number of replies in the threads, created by a member, to the total replies count in all threads.

$$Attractiveness(member_i) = \frac{card(Reply(Thread(member_i)))}{card(Reply(Thread))}$$

where  $Reply(Thread(member_i))$  – replies in the threads, created by a  $i$ -th member

$Reply(Thread)$  – total replies count in all threads.

Query for the second way of calculation of attractiveness looks as follows:

```
SELECT
  member_id,
  SUM(replies)*100.0/(SELECT
  SUM(replies) FROM Thread) as
  Attractiveness
FROM Thread
GROUP BY member_id
```

Creativeness is calculated in two ways. The first way: Creativeness is the arithmetic mean of all divisions' positive feedbacks divided by the total number of feedbacks to each post created by the member.

$$Creativeness(member_i) = \frac{\sum_j^N card(Feedback_{positive}(post_j(member_i)))}{card(Post(member_i))}$$

where  $Feedback_{positive}(post_j(member_i))$  – all positive feedbacks to the  $j$ -th post made by  $i$ -th member;

$Feedback_{total}(post_j(member_i))$  – all feedbacks to the  $j$ -th post made by  $i$ -th member;

$Post(member_i)$  – all posts created by  $i$ -th member.

Query for the first way of calculation of creativeness looks as follows:

```
SELECT
  F1.member_id,
  COUNT(*)*100.0/(SELECT COUNT(*)
  FROM Feedback as F2
  WHERE F2.member_id=F1.member_id)/
  (SELECT COUNT(*) FROM Post as P
  WHERE P.member_id=F1.member_id) as
  Creativeness
FROM Feedback as F1
WHERE F1.value=1
GROUP BY F1.member_id
```

The second way of calculating the member creativeness is: as a relation of the sum of views of all threads, created by a member, divided by the total number of all views of all forum threads.

$$Creativeness(member_i) = \frac{card(View(Thread(member_i)))}{card(View(Thread))}$$

where  $View(Thread(member_i))$  – number of views of threads, created by the  $i$ -th member,

$View(Thread)$  – total number of views of all threads.

Query for the second way of calculation of creativeness looks as follows:

```
SELECT
  member_id,
  SUM(views)*100.0/(SELECT SUM(views)
  FROM Thread) as Creativeness
FROM Thread
GROUP BY member_id
```

Results of all queries are stored in the table  $member\_feature$  that stores value characteristics of members (Fig. 4).

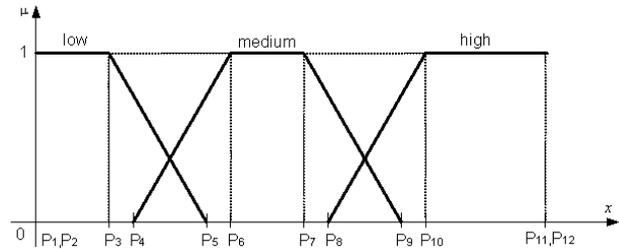


Fig. 4 – Linguistic variable

### 3.5. CALCULATION OF VALUES OF LINGUISTIC VARIABLES AND BELONGING MEASURES

Further we build linguistic variables based on the corresponding fuzzy variables in such a way.

Linguistic variable is given as a tuple  $\langle \beta, T, X, M \rangle$ , where  $\beta$  – name of variable,  $T$  – domain of variable – set of names of fuzzy variables.  $T = \{ "low", "medium", "high" \}$ ,  $X$  – domain of fuzzy variables, that describe linguistic variable;  $X = [0;1]$ ;  $M$  – set of membership measures of fuzzy variables.

The first step in calculation of values of linguistic variables is defining the interval of linguistic variable to which the member's characteristic belongs.

In the information system, for defining this interval  $BorderDefinition(feature, numeric\_value)$  function was realized. This function is based on the following query.

```
SELECT
  p1.feature_variable_id,
  p1.value as leftvalue,
  t1.value as lefttype,
  p2.value as rightvalue,
  t2.value as righttype
FROM feature_parameter as p1
INNER JOIN feature_variable as v on
  v.id = p1.feature_variable_id
INNER JOIN parameter_type as t1 on
  p1.parameter_type_id = t1.id
INNER JOIN parameter_type as t2 on
  t2.value = t1.value+1
INNER JOIN parameter as p2 on
  p2.parameter_type_id = t2.id AND
  p2.feature_variable_id =
  p1.feature_variable_id
WHERE
```

```
v.feature_id = feature AND
p1.value = (SELECT MAX(p2.value)
FROM feature_parameter as p2
WHERE p2.feature_variable_id =
p1.feature_variable_id AND p2.value
<= numeric_value)
```

This is an assisting function that helps to define fuzzy values of linguistic variables. For defining of fuzzy values of linguistic variables authors created a function *LinguisticName(feature, numeric\_value)*, which is based on these two queries:

```
INSERT INTO Measures(
feature_variable_id,
belonging_measure)
SELECT
feature_variable_id,
CASE
WHEN lefttype=1 AND
rightvalue<>leftvalue
THEN (numeric_value -
leftvalue)/(rightvalue -
leftvalue)
WHEN lefttype=2 OR
rightvalue=leftvalue
THEN 1
WHEN lefttype=3 AND
rightvalue<>leftvalue
THEN (rightvalue -
numeric_value)/(rightvalue -
leftvalue)
```

```
END as belonging_measure
FROM BorderDefinition(feature,
numeric_value)
```

```
SELECT TOP 1
feature_variable_id
FROM Measures as m
INNER JOIN feature_variable as v on
v.id = m.feature_variable_id
WHERE belonging_measure = (
SELECT MAX(belonging_measure)
FROM Measures)
ORDER BY ranking DESC
```

Next step is figuring out the belonging measure of the member to the value of the linguistic variable.

With this purpose in mind, the function *BelongingMeasure(feature, numeric\_value)* was created and it is based on the following queries

```
INSERT INTO Measures(
feature_variable_id,
belonging_measure)
SELECT
feature_variable_id,
CASE
WHEN lefttype=1 AND
rightvalue<>leftvalue
THEN (numeric_value -
leftvalue)/(rightvalue -
leftvalue)
WHEN lefttype=2 OR
rightvalue=leftvalue
THEN 1
```

```
WHEN lefttype=3 AND
rightvalue<>leftvalue
THEN (rightvalue -
numeric_value)/(rightvalue -
leftvalue)
END as belonging_measure
FROM BorderDefinition(feature,
numeric_value)
and
SELECT MAX(belonging_measure) FROM
Measures
```

If a member's characteristic falls into the interval that is covered by two values of linguistic variable, then the member will get the value whose belonging measure is greater.

Linguistic variables and belonging measures values are calculated with the help of following query.

```
SELECT
member_id,
feature_id,
LinguisticName(feature_id,
numeric_value),
BelongingMeasure (feature_id,
numeric_value)
FROM member_feature
```

Query results are stored in the table *member\_feature*.

Next step is classifying the members based on the classifying rules. Members' classification in the information system of Web-forums' members' behavior analyzing is realized on the basis of the rules that are stored in the data tables *class* and *rule* with the help of the following query

```
SELECT DISTINCT
mf1.member_id,
r1.class_id
FROM member_feature as mf1
INNER JOIN rule as r1 on
r1.feature_variable_id =
mf1.linguistic_value
WHERE r1.class_id NOT IN (
SELECT r2.class_id
FROM rule as r2
LEFT JOIN member_feature as mf2 on
mf2.linguistic_value =
r2.feature_variable_id AND
mf2.member_id = mf1.member_id
WHERE mf2.member_id IS NULL)
```

Query results are stored in the table *member*.

Classification of members will be according to the following rules:

1. If Activeness(Member)= "high" and Creativeness(Member)= "high", then Member – Activist;
2. If Activeness(Member)= "high" and Loyalty(Member)= "high", then Member – Moderator;
3. If Activeness(Member)= "low" and Reactivity(Member)= "low" and

- Creativeness(Member)= “high” and Attractiveness (Member)= “high”, than Member – Author;
4. If Creativeness (Member)= “low” and Loyalty(Member)= “low” and Reactivity(Member)=“high”, then Member – Critic;
5. If Activeness(Member)= “high” and Creativeness(Member)= “low”, then Member – Flamer.
6. If Activeness(Member)= “low” and Creativeness(Member)= “low”, then Member – Reader.

#### 4. CONCLUSION

During the information system development existing Web-forum CMS’ were analyzed, process of Web-forums’ members’ behavior analyzing was researched and overviewed, information system database scheme was developed, and algorithms of Web-forum members behavior analyzing were programmed.

Development of information system of Web-forums’ members’ behavior analyzing and classifying is very important, since Web-forums are a very popular web-service and administrating them is a hard and complicated task.

A system developed by the means of DBMS Microsoft SQL Server. It allows Web-forum owners and administrators to analyze the behavior of Web-forums’ members’ and making the right decisions in the process of moderating a community.

#### 5. REFERENCES

- [1] Kravets. R. Typical ways of web-communities development / *Kravets. R., Peleschyshyn A.M., Syerov Yu.* // *Proceedings of the International Conference on Computer Science and Information Technologies, CSIT'2006, September 28th-30th, Lviv, Ukraine, p.56–58.*
- [2] Syerov Yu.O. Simulation of behavior and classification of Web-communities participations of the base of fuzzy sets // *Transactions of National University “Lviv Polytechnics”*. – 2008. – #610 – Pp. 218-228. (in Ukrainian)
- [3] Syerov Yu.O. *Fuzzy Sets Usage for Web-Community Participants Activity Simulation* // R.B. Kravets, Yu.O. Syerov / *Transactions of Kharkiv national University of Radioelectronics “Automatic control system and automatic devices”*, issue. 141, 2007. – Pp. 113-118. (in Ukrainian)
- [4] Kruglov V. V. *Fuzzy Logic and Artificial Neural Networks: Tutorial* / Kruglov V. V., Dli M. I., Golunov R. Yu. // – Moscow: Publishing House of Physical and

- Mathematical Literature, 2001. – 224 p. – ISBN 5-94052-027-8. (in Russian)
- [5] Welser H. T. *Visualizing the Signatures of Social Roles in Online Discussion Groups* / Howard T. Welser, Eric Gleave, Danyel Fisher, Marc Smith // <http://www.cmu.edu/joss/content/articles/volume8/Welser/>
- [6] Brandes U *Exploratory Network Visualization: Simultaneous Display of Actor Status and Connections* / Ulrik Brandes, Jörg Raab, Dorothea Wagner // [Online] <http://www.cmu.edu/joss/content/articles/volume2/BrandesRaabWagner.html>
- [7] Freeman L. *Visualizing Social Networks*, / Linton C. Freeman, [Online] <http://www.cmu.edu/joss/content/articles/volume1/Freeman.html>
- [8] Peleshchyshyn A.M. *Web 2.0 – the second chance for UANet [Electronic Resource]* Online Journal of T. Shevchenko Scientific Society, 2006. – [Online]: <http://ntsh.org/uaweb2>
- [9] Pasichnyk V.V. *Web-community Efficiency Criteria Analysis on the Forums Base* / Pasichnyk V.V., Kravets R.B., Syerov Yu.O. // *Proceedings of 12<sup>th</sup> International Youth Forum “Radioelectronics and Youth in XXI century”*, 1-3 April 2008. Part 2, p. 493, Kharkiv. (in Ukrainian)



**Yury Syerov**, master of computer sciences, National University “Lviv Polytechnics” 1998-2003. Assistant at department of Information systems and networks.

Scientific interests: Web-communities, organization of effective Web-communities, web mining technologies.



**Ruslan Kravets**, Lviv Ivan Franko State University, 1990-1995.

Associated Professor at department of Information systems and networks.

Scientific interests: intelligent systems, data mining technologies, text mining and web mining, databases and

data warehouses.