An Intelligent System Based on Ontologies for Determining the Similarity of User Preferences

Oksana Oborska [1], Mykhailo Teliatynskyi [1], Dmytro Dosyn [1], Vasyl Lytvyn [1], Svitlana Kostenko [2]

[1] Lviv Polytechnic National University 12, Stepana Bandera st., Lviv, 79013, Ukraine
[2] Ivan Franko Lviv National University 1, Universytetska st., Lviv, 79001, Ukraine

Abstract
The work examines the construction of an intelligent system for searching for users with similar preferences. The system includes a series of tests and surveys to determine preferences. The central component of the system is the ontology. This ontology stores the users of the system and their answers to the tests and surveys. Based on the results of the tests and surveys, connections are made between their elements in the ontology. Based on these connections, the system makes decisions about the similarity of user preferences. The system has been developed and tested as a website. The test results are satisfactory.

Keywords
Intelligent system, ontology, information search, personal preference, personal interest.

1. Introduction

Nowadays, people use the Internet not only to search for information, but also to find new acquaintances and people with common interests. This is especially obvious in today's conditions, when the world situation is quite unstable, people are often psychologically depressed, without moral support, need advice in one or another situation, as well as help or simply communication with people with the same interests as theirs [1-4]. The development of an intelligent system that can quickly analyse and search for people with common interests is an urgent task. Such search systems combine two fundamental tasks: the automation of the determination of an individual's preferences and psychological state, and the socialisation of users [5-8].

An ontology is used to determine user preferences and to search for people with similar interests [9]. We believe that the use of ontology for such systems is the right solution, because the Internet is filled with a significant amount of data, the size of which is growing rapidly and is usually unstructured. Ontologies serve to organise knowledge and are used in areas where it is necessary to discover new facts, to reveal hidden relationships between elements. The ontology was created using the Protege editor, which has a large number of functions and supports the installation of additional plug-ins [10]. The PHP programming language was used to write the functions of the intelligent system [11].

2. Ontology construction

The following requirements are set for the functional content of the intelligent system:

- The possibility of registration and authorization of users with the input of contact data;
- The ability to edit the user profile;
- Search for users with common preferences and interests;
- Liking and saving the profile of such users;

COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20–21, 2023, Kharkiv, Ukraine
EMAIL: oksana.v.oborska@lpnu.ua (O. Oborska); <mykhailo.teliatynskyi.mknsp.2021@lpnu.ua> (M. Teliatynskyi); Dmytro.h.dosyn@lpnu.ua (D. Dosyn); vasyl.v.lytvyn@lpnu.ua (V. Lytvyn); svitlana.kostenko@lnu.edu.ua (S. Kostenko)
ORCID: 0009-0001-0825-1379 (O. Oborska); 0009-0007-2439-0401 (M. Teliatynskyi); 0000-0003-4040-4467 (D. Dosyn); 0000-0002-9676-0180 (V. Lytvyn); 0000-0003-0823-9801 (S. Kostenko)

- The opportunity to pass various surveys and tests;
- The possibility of saving the results of completed surveys;
- Providing recommendations to the user.

The knowledge base is the object that usually has to be worked with in practice. It contains knowledge from any subject area, collected by experts and recorded in the form of axioms and facts. A knowledge base recorded in a suitable computer format (usually OWL language with XML syntax) is called an ontology. Having such an ontology, specialists use special software systems (nominal arguments) to check it for feasibility, the absence of obvious errors (for example, by checking the feasibility of all atomic concepts), and also perform requests to extract knowledge logically following from those contained in the ontology. In the language of descriptive logic, logical problems correspond to these operations. The central component of the intelligent system is the ontology, which contains:

- Classes;
- Instances of classes;
- Attributes (properties) and values of these properties;
- The relationship between classes and instances of classes.

An ontology contains relationships between its elements to determine user preferences and to search for other users with similar preferences. The essence of using the ontology is that it is used to analyse the similarity of system users according to their preferences. Based on the results of the analysis, the system decides to what extent the users of the system have common interests, and these results are sent to the users. The system offers these users to get to know each other.

The main task in descriptive logic is to study the admissibility of these problems, to build algorithms for solving them, to estimate their numerical complexity, and to optimize the built algorithms.

Concepts of descriptive logic are interesting not only in themselves, but as a tool for recording information about the described subject area. This knowledge is divided into general knowledge about concepts and their relationships (so-called intentional knowledge) and knowledge about individual objects, their properties and connections with other objects (so-called extensional knowledge). The first are more stable and constant, while the second are more subject to modifications. According to this section, knowledge recorded using the language of descriptive logic is divided into:

- A set of terminological axioms or TBox;
- A set of confirmations (facts) about persons or ABox.

In practice, experts in any subject area create a terminology in which, in the form of an axiom, they fix the relationships of the main concepts (concepts and roles) that they have in this area of knowledge. After such a system of axioms is formulated, the tasks of deriving new (or, as they say, implicit) knowledge from the knowledge given explicitly in the terminology arise. One of the first tasks is to check that the terminology generally has at least one model (i.e. compatible). Next, it is verified that all atomic concepts are feasible in this terminology. If some atomic concept turns out to be unjustified, it is usually an admission that mistakes have been made in the axioms of the terminology [12-14].

To implement the task, the classes «User» and «Section» (Fig. 1) were created in the ontology, which include the following subclasses:

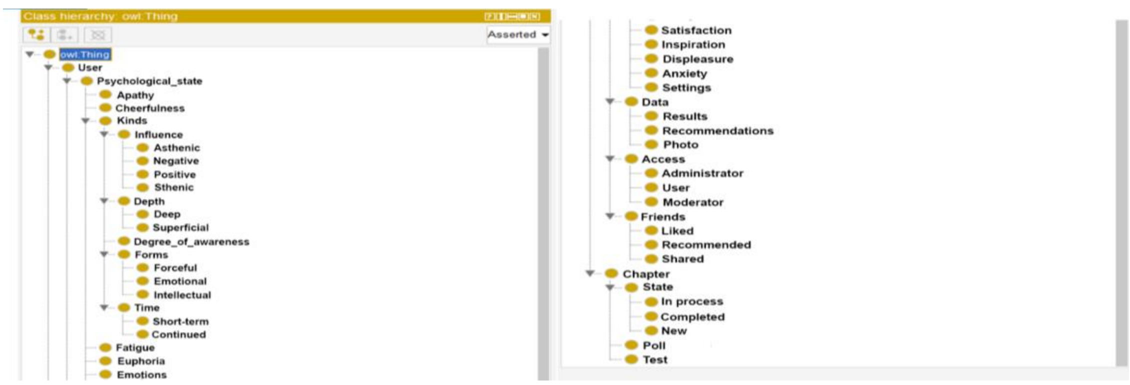
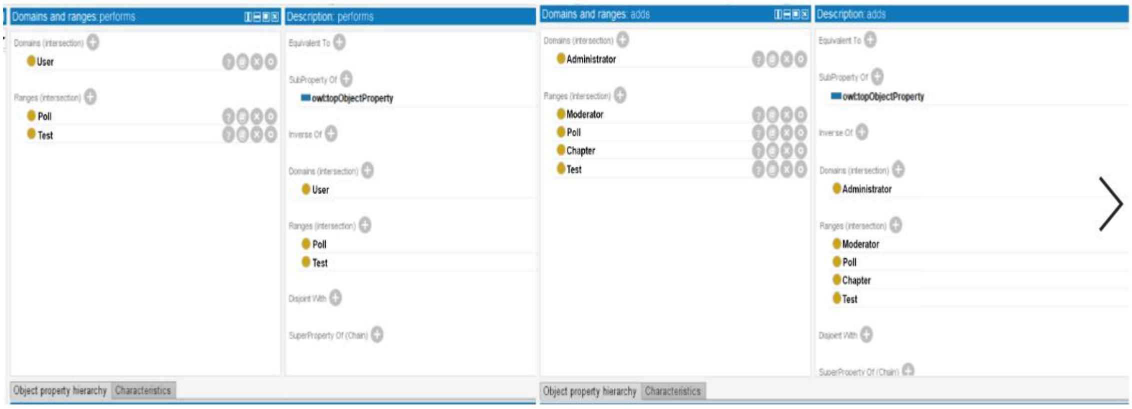


Figure 1: Created classes and subclasses

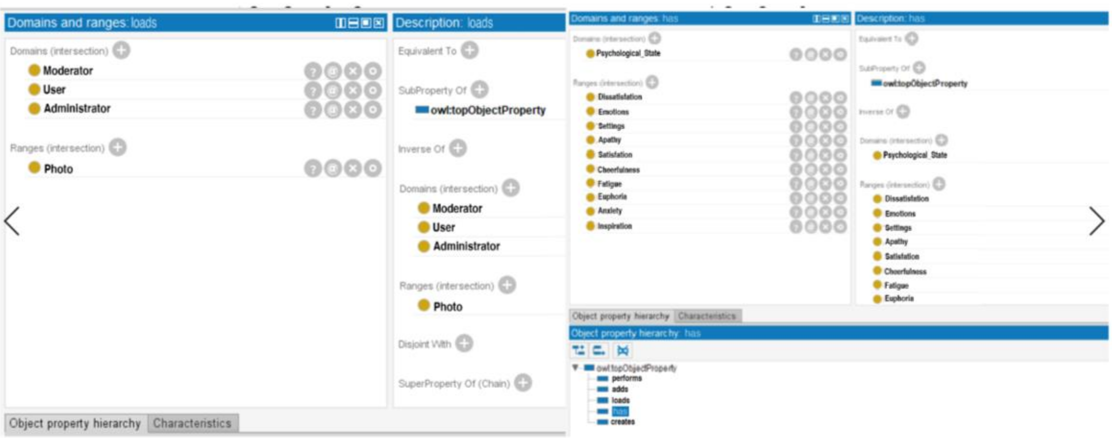
1. User – Psychological state (Species, Influence, Depth, Forms, Time, etc.), Data (Results, Recommendations, Photos), Access (Administrator, Moderator, Regular User), Friends (Favourites, Recommended, Shared).
2. Section – Status (In Process, Completed, New), Survey, Test.

The following properties were created for the given task (Fig. 2):

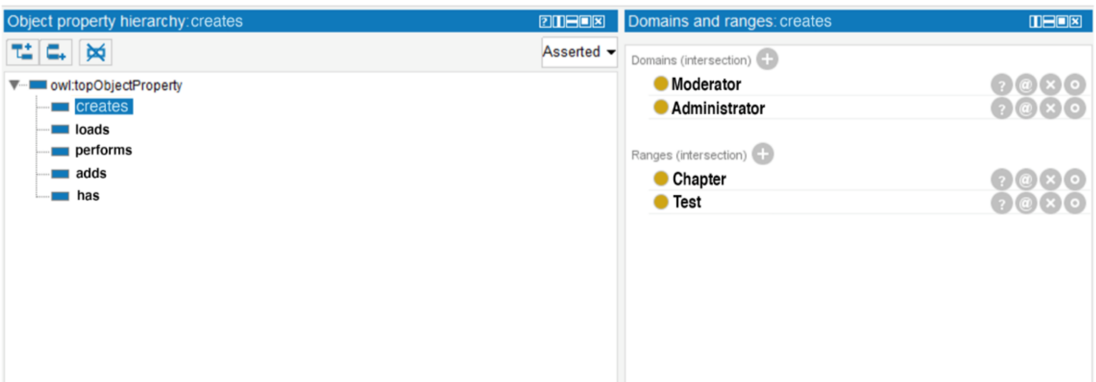
1. “creates”: domains – Moderator, Administrator; ranges – Section;
2. “loads”: domains – Administrator, Moderator, Normal user, ranges – Photo;
3. «performs»: Normal user; ranges – Survey, Test;
4. “adds”: domains – Administrator; ranges – Moderator, Survey, Test;
5. “has”: domains – Psychological state; ranges – Emotions, Satisfaction, Attitudes, Apathy, Cheerfulness, Fatigue, Dissatisfaction, Euphoria, Anxiety, Inspiration;



a) property “performs” b) property “adds”



c) property “loads” d) property “has



e) property “creates”

Figure 2: Created properties, their domains and ranges

Figures 3a-b shows some of the properties of the created ontology classes. The “date_of_creation” property – contains domains such as: Admin, Moderator, Normal_User, Results, Photos, Polls, Quiz, Shared. A data range is a data type, in this case «xsd:dateTime» , which is specifically for dates. The «name» property contains the following domains: Administrator, Moderator, Normal_user, Friends, Test, Survey, Recommended, Favorites, Shared. The data type in this case will be xsd:dateTime.

In the following, the terminology is used to derive new inclusions and equivalence concepts from existing ones.

Terminology is allowed to record general knowledge about concepts and roles. However, in addition to this, it is usually also necessary to write down knowledge about specific individuals: to which class (concept) they belong, by what relations (roles) they are related to others. This is done in a part of the knowledge base called the Individual Fact System, or Abox. To this end, in addition to a set of CN atomic concepts and a set of RN atomic roles, i.e. of names for classes and relations, a finite IN set of individual names is also introduced. Facts about individuals are of two types:

- approval of the attributes of the individual a of the concept C — written as a: C;
- the statement about the connection of two individuals a and b with the role of R — written as aRb.

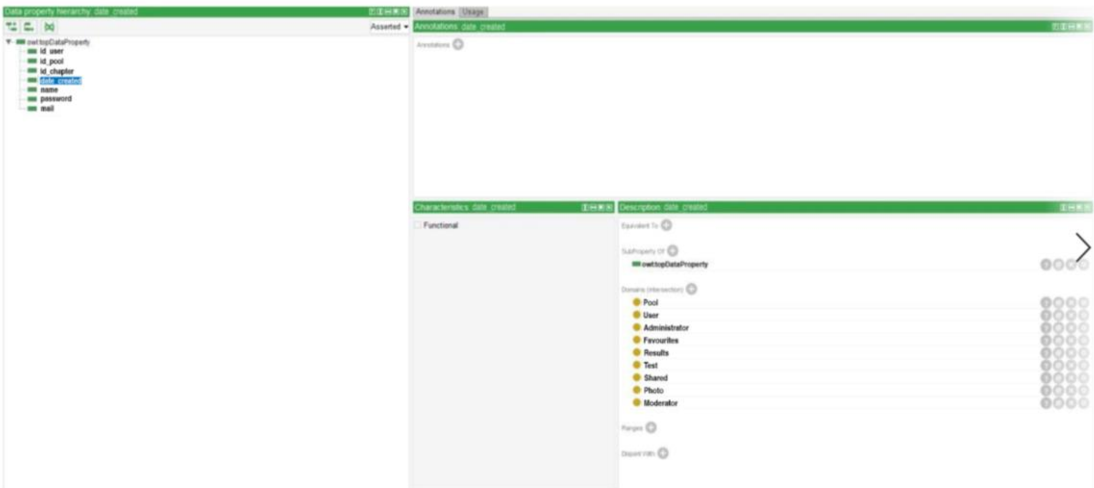


Figure 3a: "date_created" property

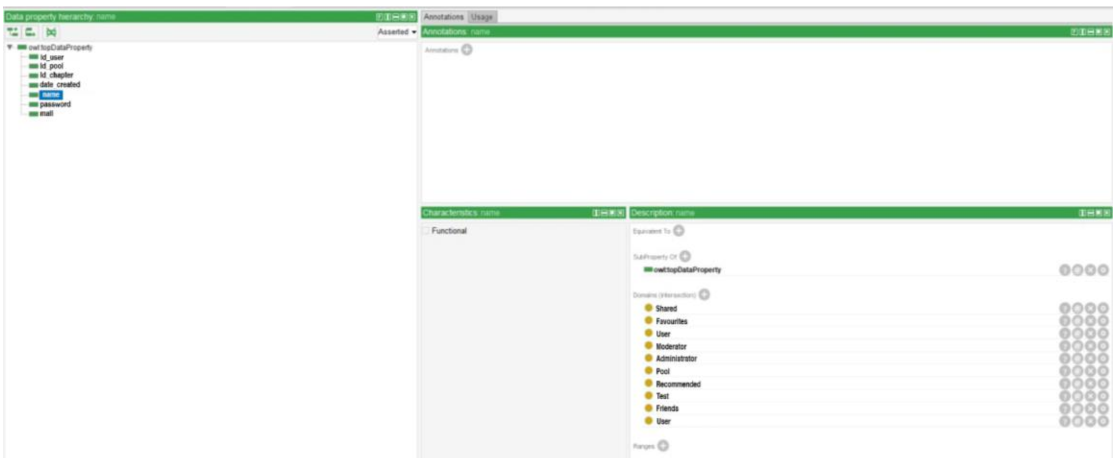


Figure 3b: "nam" property

Classes in the ontology are filled with instances. For example, an instance of the User class is an instance with the following data: name – Andriy, password – ***, e-mail – andriitest@mail.com, date of registration in the system – 2022-07-06 at 14:05:00, and also contains the property photo, which is added to the ontology. In general, the hierarchy of ontology classes looks like this (see Fig. 4).

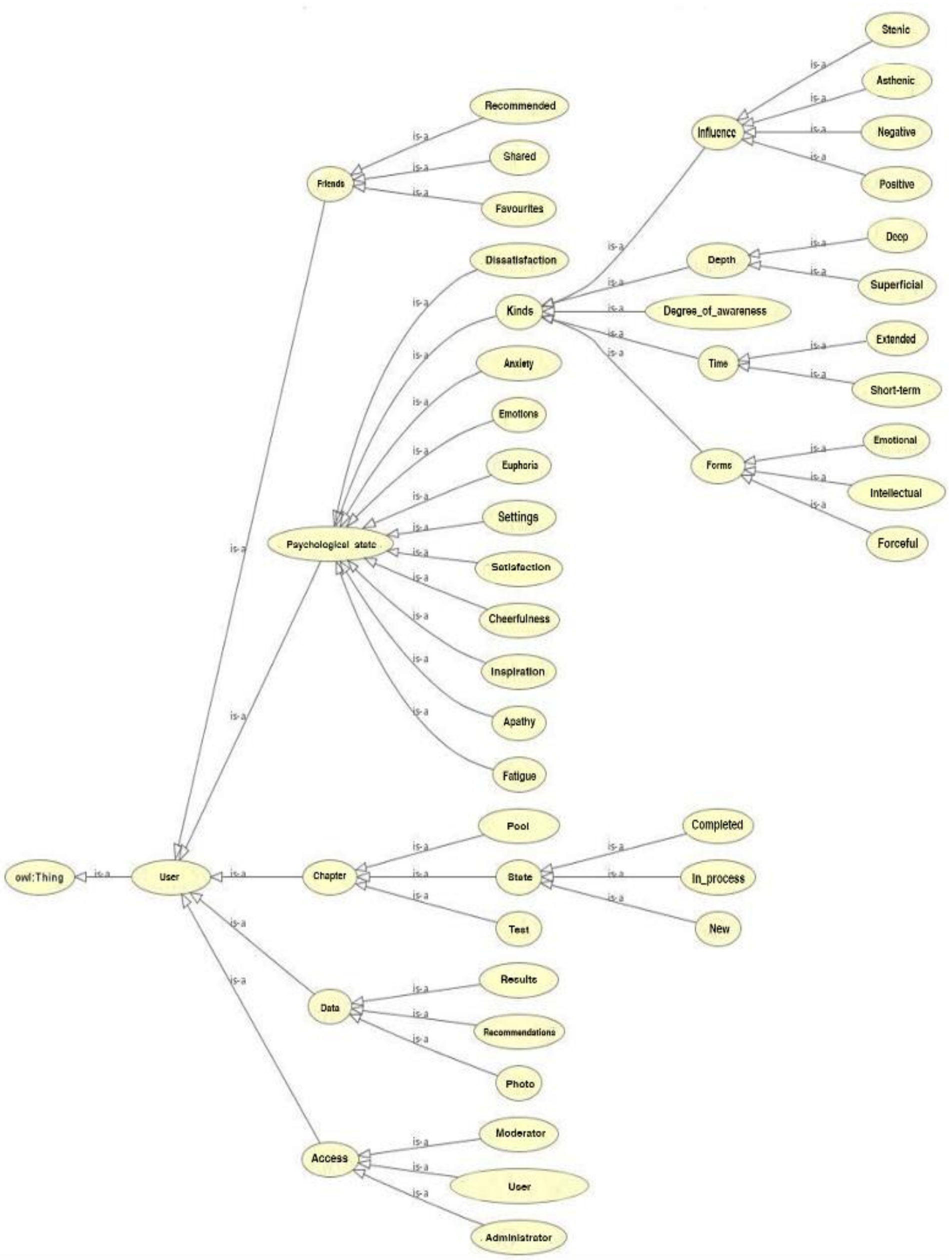


Figure 4: Hierarchy of ontology classes

The developed ontology is used by the intelligent system to perform the following functions: 1) creation of a list of users; 2) use of algorithms for list analysis; 3) calculation of the probability of similarity of users based on common interests; 4) development of interaction with the user. The intelligent system is developed in the form of a web page, which contains the following functions: authorisation; registration; storage of tokens; session monitoring; adding a photo of the user; sending requests to the server from users; displaying client errors.

In the logic of predicates of the system of facts, A corresponds to a set of closed formulas of the form P(a) and S(a, b), where P(x) is a formula with one free variable (in addition, it is not free, but is a translation of another concept), and S is a double predicate symbol (corresponds to a certain atomic role). At the same time, according to the knowledge base, this is a special case of such a well-known concept as a first-order theory, namely, it is a theory defined by a finite number of axioms and formulated in some special fragment of the language of predicate logic. Descriptive logics are constructed in such a way that the knowledge bases are permitted theories. Moreover, it is usually required that, given a theory (knowledge base) and an assertion, it is possible to efficiently (i.e., algorithmically) determine whether a given assertion is a consequence of that theory.

The developed ontology was successfully validated by the W3C RDFValidation Service [15]: "Your syntax is correct" and "Your RDF document has been successfully validated".

3. Implementation of the intelligent system and its implementation

The knowledge base is a collection of statements about a certain subject area, collected by experts in this field. In order to use this knowledge, a mechanism for extracting information from the knowledge base is needed. That's what queries are for. Let the signature $\Sigma = (CN, RN, IN)$ be given, that is, the sets of atomic concepts, roles, and individuals. To formulate queries, we introduce a new sort of symbols - a countable set of individual variables Var = {x0, x1, . . .}. Atoms (or atomic queries) are expressions of the form u: C and uRv , where C is a concept, R is a role, u, v are individuals (from IN) or variables (from Var).

For a fixed terminology T, the inclusion relation (or implication) is defined on the concepts: $T \models C \subset D$. Similarly, for a fixed RBox R, the inclusion relation between roles is defined: $R \models R \subset S$. We will extend the inclusion relation to arbitrary queries: $T \mid = q \subset p$, where T is arbitrary terminology (that is, TBox or the union of TBox and RBox , if the latter are present in the logic).

The PHP programming language was chosen for the development of the intelligent system and the PhpStorm environment [16]. The SPARQL query language and the PHP library - EasyRDF [17] were used to access the ontology. SPARQL is a language for querying ontology data presented in RDF format, as well as a protocol for transmitting these queries and responses to them. Using the EasyRDF library, a number of queries are written in the SPARQL language. An example of one of these queries is shown in Figure 5.

```
?>
<html>
<head>
    <title>EasyRdf 1st test query </title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
</head>
<body>
<h1>EasyRdf SPARQL query · </h1>

<h2>List of users:</h2>
<ul>
    <?php
    $result = $sparql->query(
     'SELECT * WHERE {'
     '   ?User rdf:type dbo:User .'
     '   ?User rdfs:label ?name .'
     '   ?User rdfs:label ?date_created .'
     '   FILTER ( lang(?label) = "en" )'.
     '} ORDER BY ?name'

    );
    foreach ($result as $row) {
        echo "<li>".link_to($row->label, $row->User)."</li>\n";
    }
    ?>
</ul>
<p>Total number of users: <?= $result->numRows() ?></p>

</body>
</html>
```

Figure 5: SPARQL query using EasyRDF

Ontology classes are filled with instances based on completed surveys and tests registered by users. These surveys and tests are published on the Internet Figure 6.

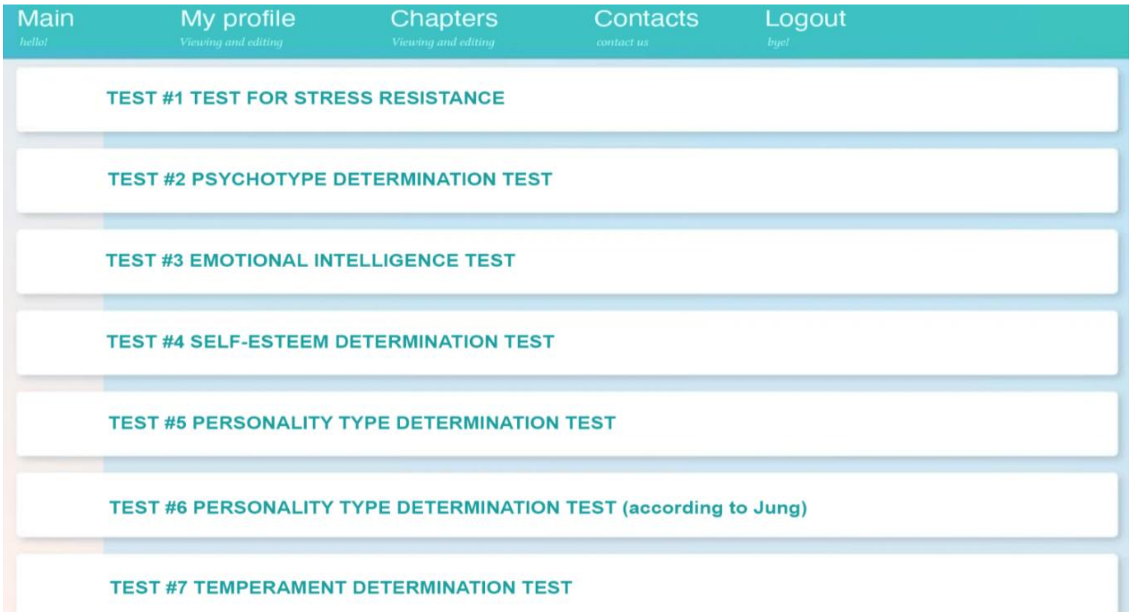


Figure 6: «Tests» section

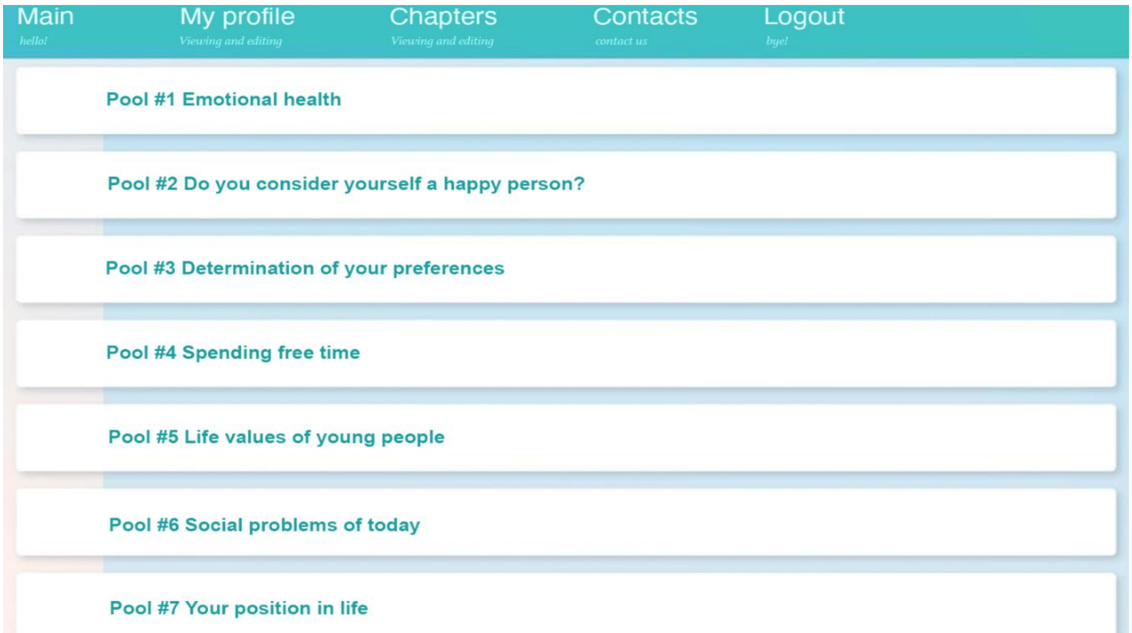


Figure 7: "Survey" section

The description of some ontology in the RDF language is a directed graph (a graph whose edges have an orientation, i.e. are arrows), the edges of which are marked by predicates, and the vertices are the subject and object of the triplet represented by this edge together with its vertices. In this case, the term RDF Store (RDF Store) is used for such graphs. In this sense, an RDF repository looks like a database structured in a specific way. Obtaining data from a relational database is usually carried out using a request written in a special SQL language (Structured Query Language - Structured Query Language), requests to databases. It can be expected that a similar approach is practiced for RDF repositories. There are indeed many different dialects of the RDF query language now. The W3 consortium has chosen one such language as a standard - SPARQL. SPARQL is a recursive acronym that stands for SPARQL Protocol and RDF Query Language. It is obvious that the name parallels the abbreviation SQL. The basic element of a SPARQL query is the triple pattern. The pattern of a triple is a triple, in some places of which there are variables instead of a subject, predicate or object. Variables are identified by identifiers beginning with the question mark "?".

SPARQL treats the input data as a directed labeled graph, which is internally expressed as triples consisting of a subject, a predicate, and an object. Accordingly, a SPARQL query consists of a set of ternary patterns in which each element (subject, predicate, and object) can be a variable (substitution pattern). Variable solutions are then found by matching patterns in the query with triples in the dataset.

Just as SQL allows users to retrieve and modify data in a relational database, SPARQL provides the same functionality for NoSQL graph databases such as GraphDB Ontotext.

In addition, a SPARQL query can also be performed on any database that can be viewed as RDF through a middleware. For example, a relational database can be queried using SPARQL using the Relational Database to RDF Mapping Software (RDB2RDF). This is what makes SPARQL such a powerful language for computation, filtering, aggregation, and subquery functions.

An example of one of the created surveys is shown in Figure 8.

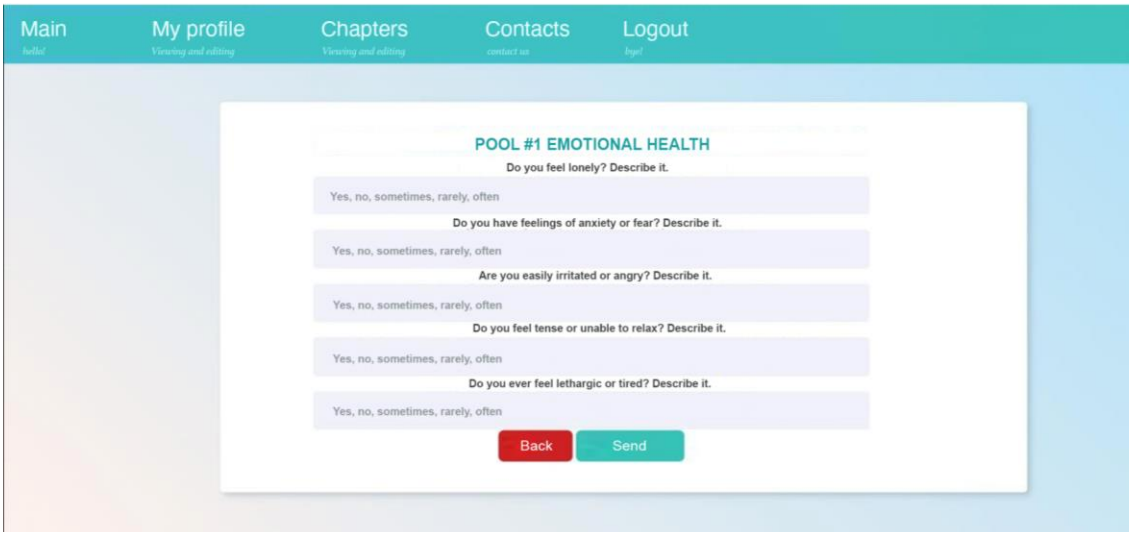


Figure 8: One of the surveys. Definition of emotional well-being

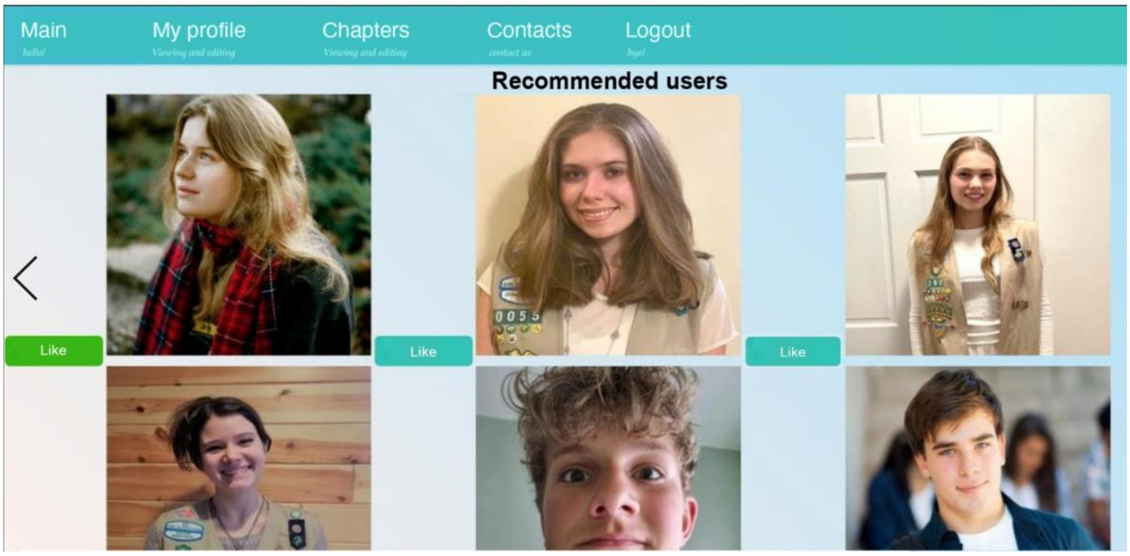


Figure 9: "Recommended" users section

If user "Bogdan" liked the profile of user "Anna", then the ontology will include it in the "Liked" subclass. Accordingly, when user "Anna" goes to the user search section, user "Bogdan" will be one of the first in the list, since the ontology has established a link between these two objects. Recommendations based on tests passed work in a similar way. Users with similar answers and resultsare recommended to each other.

The functioning of the intelligent system was tested by registering new users with activity on the Internet. The accuracy of the system is 97%.

4. Conclusions

The work examines the construction of an intelligent system for searching for users with similar preferences. The system includes a series of tests and surveys to determine preferences. The central component of the system is the ontology. This ontology stores the users of the system and their answers to the tests and surveys. Based on the results of the tests and surveys, connections are made between their elements in the ontology. Based on these connections, the system makes decisions about the similarity of user preferences. The system has been developed and tested as a website. The test results are satisfactory.

The system contains the following software modules:

- Registration and authorization of users with the possibility of password recovery via mail;
- Ability to edit profile and avatar data;
- The section on passing tests and thematic surveys;
- A section with the results of passed tests and the possibility of downloading data to a PC;
- Section with individual recommendations built by the ontology;
- Search preferences for a network user;
- Recommendations from users with similar interests and preferences.

The further improvement of the information network is represented by the increase of its knowledge base for greater informativeness, refinement of the ontology, improvement of the existing base and development of new sections. It is also planned to introduce an internal chat into the information system, so that users can communicate directly in the network.

5. References

[1] G. Alfian, M.F. Ijaz, M. Syafrudin, N. Fitriyani, J. Rhee, Customer behavioranalysis using real-time data processing: A case study of digital signage-based online stores. AsiaPacific Journal of Marketing and Logistics 2019, 31(1), 265-290. doi:10.1108/APJML - 03-2018-0088.

[2] E. Ferrara, R. Interdonato, A. Tagarelli , Online popularity and topical interests through the lensof Instagram, in: Proceedings of the 25th ACM Conference on Hypertext and Social Media, 2, 2014, pp. 24-25. doi: 10.1145/2631775.2631808.

[3] M. Salganik , Bit by bit: Social research in the digital age. Princeton University Press 2019.

[4] E. Kim, Y. Sung, H. Kang, Brand followers' retweeting behavior on Twitter: How brand relationship influences brand electronic word-of-mouth, Computers in Human Behavior 38(8) (2014) 18-25. doi: 10.1016/j.chb.2014.04.020.

[5] C. Kudeshia, P. Sikdar, A. Mittal, Spreading love through fan page liking: A perspective on small-scale entrepreneurs, Computers in Human Behavior 8(19) (2016) 257-270. doi: 10.1016/j.chb.2015.08.003.

[6] Handbook of Research on the Global Impacts and Roles of Immersive Media 2019.- 131- ISBN13: 978-1-79982-433-6

[7] Ontology-based Applications - url: https://www.semanticarts.com/ontology-based-applications.

[8] Protégé-OWL: Creating Ontology-Driven Reasoning Applications with the Web Ontology Language - url: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1560433/#abstract-1title

[9] A PHP library designed to make it easy to consume and produce RDF.- url: https://www.easyrdf.org.

[10] W3CRDFValidation Service - url: https://www.w3.org/RDF/Validator/rdfval

[11] F. Baader, I. Horrocks, and U. Sattler. Description Logics as Ontology Languages for the Semantic Web. Lecture Notes in Artificial Intelligence, Volume 2605. Springer-Verlag, 2005.

[12] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F.Patel-Schneider, Eds. The Description Logic Handbook. Cambridge University Press, 2003.

[13] P. Blackburn, J. van Benthem, F.Wolter, editors. Handbook of Modal Logic. Studies in Logic

and Practical Reasoning, Volume 3, Elsevier, 2007.
[14] What is SPARQL - A Complete Tutorial Guide - url: https://mindmajix.com/sparql- tutorial.
[15] Resource Description Framework (RDF) - url: https://www.techtarget.com/searchapparchitecture/definition/Resource-Description-Framework - RDF
[16] OWLVIZ Display problems - url: https://github.com/protegeproject/owlviz/issues/19
[17] Allemang D. Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL James Hendler - Morgan & Claypool Publishers, 2008.