Computer Science and Information Systems 19(1):229–250 https://doi.org/10.2298/CSIS210207050M

Semantic Web Based Platform for the Harmonization of Teacher Education Curricula

Milinko Mandić

Faculty of Education, Podgorička 6,
25000 Sombor, Serbia
milinko.mandic@pef.uns.ac.rs

Abstract. This paper describes a developed semi-automatic software platform for the harmonization of the informatics curricula at all levels of education. The applied algorithms for matching ontologies are described in detail, as well as the principle of mapping informatics curricula to ontological models. The model of the selected informatics teacher education curriculum from the Republic of Serbia was created and compared to the model of the reference informatics teacher education curriculum using a software platform. The analysis of the results includes a comparison with the data obtained for other possible pairs of the created input ontological models (the secondary school ACM K12 model and the reference model, the secondary school model and the model of the selected curriculum). The research presented in this paper indicates that it is necessary to consider the improvement of teacher education curriculum as well as the application of new matching techniques.

Keywords: ontology, alignment, matching, teacher education curriculum, informatics.

1. Introduction

The rapid changes in the field of computer science (CS) require the constant improvement of the CS curricula. Primary, secondary and higher education CS curricula must be mutually aligned, but also have to be harmonized with the development of CS field. Therefore, it is necessary to present the curriculum in such a form that is computer interpretable and easy to change. Also, it is important to facilitate the determination of the curricula harmonization of different levels of education. This could be achieved by applying a software platform that would point out the missing aspects in the curriculum, provide statistical data, the possibility of improving the curriculum model and the like.

The rest of the paper is organized as follows. Section 2 shows related work. Section 3 presents the ontological models of the reference and chosen informatics teacher education curricula. Section 4 describes the architecture of our platform for curricula harmonization and ontology matching methods, applied in the platform. Section 5 gives discussion of the results following the application of the presented software to created ontological models. Section 6 contains concluding considerations, limits of the developed platform and future research aims.

2. Related work

Seitz [1] states that researchers often examined the alignments between the intended and the assessed curricula and between the enacted and the assessed curricula. Bay et al. [2] present the research with the aim to establish the factors that affect curriculum alignment. In the paper, the curriculum alignment is defined as “the compatibility between a country's centralized curriculum determined by the ministry of education and what teachers do during the teaching process”. Digital curriculum mapping tool, presented in [3], was created mainly to facilitate „processes of improving curriculum alignment and visibility of learning trajectories for teachers and students”. In this paper, the curriculum alignment is interpreted through the „dynamics” between the program structure and the student’s learning. The tool provides students, teachers and curriculum evaluators with a quick and easy insight into how and when the acquisition of certain curriculum skills and knowledge is planned. Moreover, the purpose of the created curriculum mapping tool is to enable teachers to better understand the curriculum and the position of the course they teach within a predefined learning trajectory. The developed digital mapping tool is especially important in the accreditation process, as it enables easy access to the content related to the visualized learning trajectory, providing information about thematic areas which the educational institutions consider important. In [4] the author analyzes the possibilities of electronic curriculum mapping system e-CMS for organizing curriculum alignment initiatives. The proposed system can be applied to both internal and external alignments. The internal alignment refers to determining the compliance of the three elements of a course, teaching and learning activities, assessments, and objectives. The external alignment is used to check the consistency of the courses with one another. A research, shown in [5], presents constructive alignment with a cross-institutional study from two Australian universities. The paper emphasizes the importance of two approaches: the top-down institutional alignment implementation at one university and the bottom–up approach within the other. The top-down approach starts with a corporate strategy (higher education institution) and follows a series of sequential steps ending with individual student assessment learning items. The bottom–up approach implies the reverse direction. It can be seen from the cited literature that all analyzed papers and presented tools have in common the investigation of the curriculum alignment by determining the extent to which the teaching is synchronized with the predefined curriculum. They do not deal with the harmonization of curricula with external recommendations or other curricula.

A number of papers in current literature [6-9] testifies to the suitability of using ontologies for curriculum representation. Ontologies ensure the presentation of the curricula so that they can be interpreted by machines. The authors in [8] state that the ontological representation of curricula provides easier curricula alignment. In [9], Electrical Engineering Curriculum was represented through ontologies. The preliminary research, by using the model of a semi-automated Academic Tutor, indicated that the formal representation of the curriculum's knowledge could be shared and reused in the field of education and engineering. The authors in [10] developed a new classroom teaching model driven by the curriculum ontology. It was used in the creation of a teaching plan and verified in the course of E -Commerce. Ref. [11] states that some of the benefits of using ontologies are: sharing common understanding of the information structure, „facilitating reuse of domain knowledge”, analyzing domain knowledge. A model of a semi-automatic build of the intelligent curricula based on the existing educational resources in digital format (such as digital books, web/based tutorials or curricula), was proposed in [12].

Contemporary literature presents many ontology alignment systems that use numerous methods and techniques for ontology matching. Some of them are shown in [13-15].

From the literature review, it can be concluded that there are a number of papers dealing with curriculum alignment, curriculum ontological representation and ontology alignment. However, to our knowledge, only curricula synchronization platform, presented in [16][17], is made on the basis of ontology alignment methods. The platform for the harmonization of teacher education curriculum software, presented in this paper, uses a different input ontological model compared to those presented in [16][17]. Also, the ontology matching process uses (new) algorithms adapted to the specificity of teacher education curriculum comparisons. This is especially true of terminological and taxonomic structural algorithms.

3. Teachers' Curricula Ontological Models

The motivation for the development of the platform shown in this paper is to establish whether the graduates of informatics teacher preparation programs are competent to implement teaching in primary and secondary schools in accordance with contemporary international standards and recommendations. Therefore, the main upper class of the teachers' ontological models is the *Competence* class. Different definitions of competence are summarized in [18]. From [18] it can be seen that competence almost always implies acquiring knowledge and skills. Thus, direct subclasses of the *Competence* class are *Knowledge* and *Skills* classes that are mutually connected via *hasKnowledge/hasSkills* object properties. According to the relevant literature [19-20] (revised) Bloom's taxonomy is particularly suitable for use in computer science field. Therefore, the *Skills* class is modelled based on cognitive domain of the Revised Bloom's taxonomy, i.e., the following classes: *Evaluate, Create, Analyze, Apply, Remember-understand* are subclasses of the *Skills* class.

Computer Science Teachers Association (CSTA), the organization which promotes and supports CS teaching [21], suggests models for educating teachers relying upon the ACM model K12 curriculum of computer science. The paper suggests that any programme for preparing CS teachers must include the four main components: academic requirements in the field of computer science; academic requirements in the field of education; methodological (a methods course) and field experience; general pedagogical knowledge. National Council for Accreditation of Teacher Education (NCATE), the accreditation body in the USA for the accreditation of study programmes that educate future teachers, has developed (since 1990) a series of standards for preparing secondary CS teachers by promoting programmes based on K12 model curriculum. The proposal, shown in [22], lists knowledge and skills that CS teachers should have.

A detailed insight into the CS (informatics) teachers' curricula around the world (USA, Serbia, Israel, Estonia, Turkey, Austria, Germany, Scotland) as well as the analysis of reference CSTA/NCATE standards [21][22] and current literature [23] shows that informatics teacher curricula should cover the following content fields:

general knowledge, general educational and pedagogical knowledge, informatics domain knowledge, knowledge of teaching practice, knowledge of informatics teaching methods. These fields are mapped onto appropriate classes and are modelled as *Knowledge* subclasses (Fig. 1).

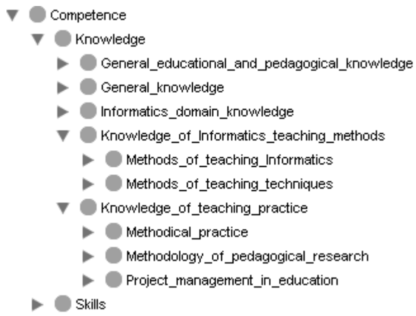

Fig. 1. Upper hierarchical structure of the *Knowledge_of_teaching_practice* class

The developed alignment software is applied in this work in order to compare ontological models of the reference to the selected curricula. NCATE/CSTA standards and curricula from different countries are mapped to the „reference" teacher education model. The study program for informatics and technology teachers is mapped to the "selected" teacher education model. The procedure of creating the *Knowledge* subclasses implied that the courses (or content fields) were mapped to direct subclasses of one of the five general knowledge areas (shown in fig. 1), while the topics contained in the courses were represented as lower subclasses. An additional description of the topics (usually shown in parentheses), if any, is mapped to the classes' labels. Figure 1 presents a part of the hierarchical structure of the *Knowledge_of_Informatics_teaching_methods* class in the selected curriculum model. The *Skills* class structure was created as follows: learning outcomes were classified based on the cognitive domain of the Revised Bloom's taxonomy and represented by lower *Skills* subclasses.

Chosen teacher education curricula model is based on the study program "Informatics and techniques in education" at the Technical faculty, Zrenjanin [24]. The method of mapping the curriculum into the ontological model is analogous to the procedure described in [17]. Also, the ontological models are given in owl format at [25].

4. Semantic Web Based Platform for Curricula Synchronization

The developed software platform is based on ontology alignment that is, in this paper, interpreted as a „set of correspondences" [26] between two ontological models of teacher education curricula curricula (O_1 and O_2). Object and datatype properties are predefined and the same in both ontologies, while instances are not included in the models. Therefore, the "set of correspondences" consists only of the classes' pairs (C_{i1},C_{j2}), similarity values (confidence degree - conf_i) between the classes and relations among them (Equation 1). Possible relations between classes are: the superclass, the subclass and the equivalence.

$$\text{Alignment}(O_1, O_2) = \begin{cases} \left(C_{i1}, C_{j2}, \text{conf}_i, \text{relation}_i\right) \mid \\ C_{i1} \in O_1, C_{j2} \in O_2, \text{conf}_i \in [0,1], \\ \text{relation}_i \in \{=, \subseteq, \supseteq\} \end{cases} \quad \textbf{(1)}$$

The presented ontology alignment system predicts “one to one” and “one to more” (1:N) relationships. This means that a set of classes from one ontology can be the subclasses of a class from the other ontology. The ontology alignment is done in several steps. In the first phase, terminological similarity is determined. Similarity matrix, obtained by applying terminological matcher, represents the input for all following matchers. In the second phase, ontological matching methods are applied to determine taxonomic structural similarity, relational similarity and one-to-many similarities, respectively. The results of the application of each matcher represent the input for the next. The developed software platform provides the user with a change of the results after all matching steps, apart from after the terminological one.

A similarity matrix is created after each matching phase. It consists of the similarities of all possible the classes’ pairs of compared ontologies. The aim of using matchers is to establish “the best matched classes” i.e. to find the pairs of classes (for “1 to 1” relation) that are mostly alike (closest).

A determination of the best matched classes from the similarity matrix as well as a detailed description of each matcher is given in the following sections.

4.1. Determination of the Best Matched Classes

The problem of matching is well studied in literature dealing with graph theory [27], according to which it is possible to apply several criteria for determining the best matched pairs: maximum cardinality, maximum total "weight" and "stable marriage" Matching has maximum cardinality if it has the largest number of mappings (paired fields); matching has a maximum total weight if the sum of the weights of its mapping is the greatest; a "stable marriage" requires that there are no such combinations of paired fields (x, y) and (x1, y1) so that x more "prefers" y1 than y and y1 "prefers" x more than x1. According to [27], Greedy's choice for the matching of entities with cardinality of 1:1 can be considered as "monogamous" version of the "perfectionist egalitarian polygamy" selection metric which, according to the empirical results shown in [28] gives the best results in the matching scheme. In this paper, Greedy selection algorithm is used for determining the best classes’ pairs, because this method is frequently used in the ontology alignment systems like [29] and [30]. Also, authors in [31] state that, for example, a matching that maximizes the sum of the similarities of the selected pairs, is not an "optimal" solution for the problem of ontology alignment. As a reason for this claim, in [31] it is stated that the goal of an ontology alignment is to maximize the number of correct pairs and minimize the number of incorrect pairs. Therefore, in the context of ontology alignment consideration (assuming that the value of the similarity is directly related to the likelihood that the matching is true), selecting a pair with a high similarity (for example, over 90%) may be more correct than selecting two pairs with an average similarity value (50-60%).

The Greedy method [27], applied to a two-dimensional similarity matrix, may be described as follows [26].

1. Selecting a pair of entities $e_{m1} \in O_1$ and $e_{n2} \in O_2$ which has the highest similarity value of all entity pairs.
2. "Removing" rows and columns containing e_{m1} and e_{n2} so that e_{m1} cannot be paired with any $e_{j2} \in O_2$, $j \neq n$, and e_{n2} cannot be paired with any $e_{i1} \in O_1$, $i \neq m$.
3. Finding the greatest similarity between the remaining pairs of entities.
4. Repeating the process until one value in the similarity matrix remains.

In the system described in this paper, the obtained pairs of entities become the "best paired" if they are greater than the given threshold. Figure 2 shows an example of determining the paired entities by using the described method, with the threshold value of 0.5.

e_{11}	0.9	0.2	0.4
e_{21}	0.85	0.5	0.6
e_{31}	0.4	0.8	0.5

Fig 2. Example of determining the best pairs from the 2D similarity matrix

In the first step, the greatest possible similarity contained in the matrix is selected; that is, in the example from the image, 0.9, and the first pair of paired entities is: $\{e_{11}, e_{12}\}$. In the next step, the pair $\{e_{21}, e_{12}\}$ will not be selected, although their similarity is 0.85, since e_{12} has already been matched. The next greatest similarity among the remaining unpaired entities is, then, 0.8, therefore $\{e_{31}, e_{22}\}$ is the next best matched pair. In the last step $\{e_{21}, e_{32}\}$ are matched.

4.2. Terminological Similarity

Terminological similarity is determined using string and linguistic based method. String tokenization including string normalization methods (identification of numbers, special characters, blank spaces, uppercase to lowercase conversion, removing stop words etc.) precedes the establishing of string similarities. In this phase, strings contained in local names and class labels are taken into account. The English version of the WordNet lexical database is used for morphological linguistic normalization.

The similarity of tokens contained in the local classes' names is obtained using Lin's "information-theoretic" method [32] if both tokens are in the WordNet database. If at least one of the tokens is not in the WordNet dictionary, the similarity of the tokens is determined using the Jaro Winkler method [33], [34]. S_{ln} list, consisting of similarities of the "the best matched pairs" of tokens, is gained by applying the Greedy selection method to the matrix comprising the similarities of all tokens of the C_{i1} class with all tokens of the C_{j2} class. The total similarity of the local names for the two classes $s_{ln}(C_{i1}, C_{j2})$ is then calculated. A slightly different way is applied depending on whether the classes are subclasses of the *Knowledge* class or of the *Skills* class.

Skills subclasses represent skills/outcomes that are often described by free text. The difference in the number of words contained in the outcomes can significantly affect the different meaning of the outcome. Therefore, when calculating the similarity of the local names of the *Skills* subclasses, the number of tokens should be taken into account. Thus, $s_{ln}(C_{i1}, C_{j2})$ is calculated as follows.

$$s_{ln}(C_{i1}, C_{j2}) = \frac{2 \cdot \sum_{i=o}^{m} S_{ln}(i)}{|tok_{i1}| + |tok_{j2}|} \text{ ; } |tok_{ik}| \text{- \# of tokens in local name of } C_{ik}; \qquad (2)$$

$$\text{m-dimension of } S_{ln};\ C_{ik} \text{is Skills subclass}$$

It can be seen from the above formula that the similarity between the classes, described with the different number of words (tokens), is reduced.

On the other hand, for the *Knowledge* subclasses, which are usually described by a smaller number of tokens and which represent the names of topics/thematic areas, it has been experimentally shown that more accurate results are obtained if a different principle of calculating the similarity is applied. The principle of determining $s_{ln}(C_{i1}, C_{j2})$ of the *Knowledge* subclasses depends on the ratio of the difference in the number of tokens $\left| |tok_{i1}| - |tok_{j2}| \right|$ and the minimum number of tokens $min(|tok_{i1}|, |tok_{j2}|)$. If the difference in the number of tokens is not less than the minimum number of tokens, the above formula (2) is applied. Otherwise, the total similarity of the local names of the two classes $s_{ln}(C_{i1}, C_{j2})$ is obtained as the average value of the elements of the list S_{ln}.

The similarity of the classes' labels $s_{lb}(C_{i1}, C_{j2})$, and the similarity between the local name of the class of one ontology and the label of the class of the compared ontology $s_{lnlb}(C_{i1}, C_{j2})$ and, inversely, $s_{lbln}(C_{i1}, C_{j2})$ is calculated in an analogous way. The total terminological similarity for classes $s_{term}(C_{i1}, C_{j2})$ is:

$$s_{term}(C_{i1}, C_{j2}) = \max(s_{ln}(C_{i1}, C_{j2}), s_{lb}(C_{i1}, C_{j2}), s_{lnlb}(C_{i1}, C_{j2}), s_{lbln}(C_{i1}, C_{j2})) \quad (3)$$

4.3. Taxonomic Structural Similarity

Taxonomic structural similarity includes the following stages: the determination of a parent similarity, the determination of the similarities of leaf classes and the determination of similarities of leaf classes belonging to the unmatched classes' structures. Since *Skills* part of the ontological hierarchy is less structured (classes representing a cognitive domain of Bloom taxonomy mostly have direct subclasses only), the taxonomic structural similarity is established only for the *Knowledge* subclasses. The classes in the upper classes' structure that are the same in both ontologies (like: *Knowledge*, *Competence*, *Informatics_domain_knowledge*, *General_knowledge*, etc.) are not considered in obtaining the taxonomic structural similarity.

Parent Classes' Similarity

In the first step, only the parent classes (classes that have subclasses) are compared. Parent classes' similarity is calculated based on the terminological similarity of the compared classes, the similarities of all superclasses (if they exist) and the similarities of all subclasses. Thereby, considered superclasses (parents) and subclasses (children) include direct and indirect classes in a parent/child relation. Thus, the similarity of the superclasses of the classes C_{i1} and C_{j2} $s^{sup}(C_{i1}, C_{j2})$ is determined as follows

```
/* Let A_ij be a class of an ontology| A_k1 ∈ O_1 and A_l2 ∈ O_2
if ∄A_k1|C_i1 ⊆ A_k1 or ∄A_l2| C_j2 ⊆ A_l2 then
          s^sup(C_i1, C_j2) is not taken into account
else
     Let C_i1 ⊆ {A_k1}, k=1,n; n≥1 and C_j2 ⊆ {A_l2}, l=1,m; m≥1
           for k = 1 to n
                for l = 1 to m
/* the values of the similarity of classes from the set
{A_11, A_21… A_n1} with classes from {A_12, A_22… A_m2} become
elements of matrix with n rows and m columns
          matrix[k][l] = s_term(A_k1, A_l2)
/* By applying Greedy selection method on the matrix list
of the best matched superclasses' pairs S^sup is obtained
S^sup = Greedy_Selection_Method (matrix)
```

$$s^{sup}(C_{i1}, C_{j2}) = \frac{\sum_{i=o}^{m} S^{sup}(i)}{m}, \quad m = \text{size of } S^{sup}$$

The similarity of the subclasses $s^{sub}(C_{i1}, C_{j2})$ is calculated in an analogous way [16]. The total similarity $s_{parent}(C_{i1}, C_{j2})$ of classes C_{i1} and C_{j2} is calculated as follows

```
If ∃A_k1|A_k1 ⊆ C_i1 and ∃A_l2|A_l2 ⊆ C_j2 then
```

$$s_{parent}(C_{i1}, C_{j2}) = \frac{s^{sup}(C_{i1}, C_{j2}) + s^{sub}(C_{i1}, C_{j2}) + s_{term}(C_{i1}, C_{j2})}{n};$$

```
      n=2 in case when ∄A_k1|C_1 ⊆ A_k1 or ∄A_l2|C_j2 ⊆ A_l2
(when s^sup(C_i1, C_j2) is not taken into account.
Otherwise, n=3
else
```

$$s_{parent}(C_{i1}, C_{j2}) = 0$$

The resulting similarity matrix S_{parent} contains calculated similarities between all *Knowledge* subclasses (not including predefined classes) from ontology O_1 with all *Knowledge* subclasses of the O_2 ontology. The best matched classes from S_{parent} are achieved using Greedy selection algorithm with the given threshold.

Leaf Classes' Similarity

In the next phase, only the similarities of leaf classes are calculated. List (leaf) classes are classes that do not contain their own subclasses. The similarity values calculated by terminological matcher are allocated to the pairs of leaf classes if some of their parent classes are matched by using previous matcher for determining parent similarity. If this is not the case, or if one of the compared classes has subclasses, their similarity is $s_{parent}(C_{i1}, C_{j2})$. In this stage, leaf classes and classes that have only leaf subclasses are also considered (by using the same principle of assigning similarity values for leaf classes). The reason for extending the algorithm to these classes is the possibility that a topics/thematic area in one of the curricula is described in more detail with leaf subclasses, although it is equivalent to the compared leaf class. This exception is used for the superclass/subclass relations described in the 1:N algorithm.

Unmatched Parents' Leaf Classes' Similarity

The classes belonging to the "unmatched classes structure" take into account the last step of the calculation of taxonomic structural similarity. The motivation for the introduction of this algorithm is the possibility that related thematic areas are represented by a different number of classes' structures but at the same hierarchical level. For example, thematic areas/courses related to the study of programming can be represented by two upper classes structure in one ontology (e.g. *Programming_languages*, *Object_oriented_ programming*), while in the other ontology related (or even the same) programming topics can be mapped onto subclasses of three or more upper classes' structure (e.g. *Programming_languages*, *Introduction_to_programming*, *Object_oriented_ programming*). In this case, it is possible that some related (or even the same) topics are mapped onto subclasses that would not be matched since their superclasses are not paired by the parent 1:1 matcher.

This algorithm uses disjoint property from the OWL (e.g. listed superclasses that represent programming concepts would not be signed as disjoint) in the following way [16]:

/* Let A_{leaf} be a list of matched classes obtained by a matcher that determines the similarity of the leaf classes of matched parents.
/* Let the following apply:
$\{\{A_{11}, A_{12}\}...\{A_{n1}, A_{n2}\}\} \in A_{leaf}, C_{i1} \subseteq \{A_{11}...A_{n1}\}, C_{j2} \subseteq \{B_{12}...B_{m2}\},$
$A_{k2} \in \{A_{12}...A_{n2}\}, B_{k2} \in \{B_{12}...B_{m2}\}$
If C_{i1} and C_{j2} are unmatched leaf classes and $\nexists A_{k2}$, B_{k2} defined as disjoint classes and
$\nexists\{A_{11}, B_{k2}\} \mid \{A_{11}, B_{k2}\} \in A_{leaf}, A_{11} \in \{A_{11}...A_{n1}\}, B_{k2} \in \{B_{12}...B_{m2}\}$
then

$$s_{disj}(C_{i1}, C_{j2}) = s_{term}(C_{i1}, C_{j2})$$

else

$$s_{disj}(C_{i1}, C_{j2}) = s_{leaf}(C_{i1}, C_{j2})$$

4.4. Relational Similarity

The relational similarity is calculated only between the *Skills* subclasses as follows: if the compared classes C_{i1} and C_{j2} are connected via *hasKnowledge* object property to the

Knowledge subclasses that belong to paired classes structures (i.e. there is at least one pair of *Knowledge* subclasses $\{B_{k1}, B_{m2}\}$ obtained by the taxonomic matcher so that B_{k1} is related to C_{i1} and B_{m2} to C_{j2}), then pair $\{C_{i1}, C_{j2}\}$ gets a similarity value calculated by the terminological matcher.

4.5. 1:N matcher

1:N matcher pairs a class of one ontology with leaf classes of another ontology in "superclass/subclasss" relation, provided that the observed class of one ontology is matched with the parent class of the leaf classes of the other. It is based on the special case of the leaf matcher where a list class from one ontology and a class that has only leaf classes from another ontology are matched. The method for obtaining subclass relation can be described in the following way.

/* Let A_{rel} be a list of the matched classes calculated by a previous relational matcher
If $\{C_{i1}, C_{j2}\} \in A_{rel}$ and $\exists A_{11} | A_{11} \subseteq C_{i1}$ and $\nexists A_{k2} | A_{k2} \subseteq C_{j2}$ then
If $\nexists\{A_{11}, A_{m2}\} | \{A_{11}, A_{m2}\} \in A_{rel}$, $A_{11} \in O_1$, $A_{m2} \in O_2$, $A_{11} \in \{A_{11} \ldots A_{n1}\}$, $\{A_{11} \ldots A_{n1}\} \subseteq C_{i1}$, n ≥ 1 then $C_{j2} \supseteq \{A_{12} \ldots A_{n1}\}$
/*$\{A_{12} \ldots A_{n1}\}$ from ontology O_1 are subclasses of the C_{j2} class from O_2 ontology

Superclass relation is obtained in an analogous manner.

4.6. Graphical User Interface

The graphical user interface (GUI) of the developed software platform gives an overview of the ontologies' hierarchical structure, the information about the classes and the matching results. The user can choose the input ontological models and the alignment level (the alignment of secondary school and teacher education curricula models or the alignment of teacher education curricula models). The system allows the user to enter the threshold value as well.

The GUI shows the opened ontological models in a tree structure. The classes' information is shown in the panels on the left side (Figure 3). The class' information contain related comments (if they are entered), a label, an associated class (by *hasKnowledge/hasSkill* object property) and the class to which it is matched. The results' statistics is shown in a separate tab (Figure 4).

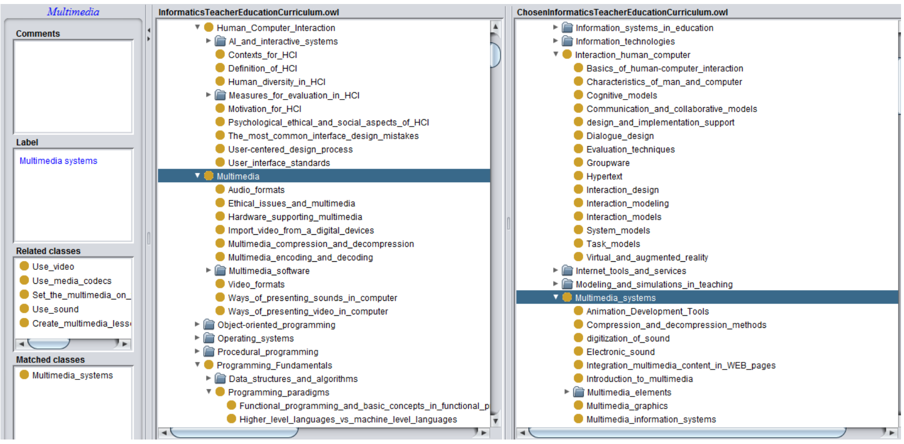

Fig. 3. A part of the classes' structure of the teacher education curricula models.

After the application of each matcher type, the tables display the matching results in the "Alignment output" tab (Figure 4). The tables contain paired classes, the relation type (superclass, subclass, equivalence) and the similarity value. When using a relational matcher, the table gets an additional column ("Bloom") that reflects the consistency of the cognitive domain of Bloom's taxonomy.

Since the local names of some classes (especially *Skills* subclasses) are the result of free text mapping (contained in learning outcomes), it was necessary to ensure that the system is semi-automatic. Hence, the GUI allows the user to improve system accuracy. Matching results can be changed in "Ontologies" tab (by using drop down menu) and "Alignment output" tab (by choosing the pairs of classes to be matched and by entering the similarity values as well as relation type). The equivalence relation is the only possible type of a relation for all matchers except for the last one. Therefore, the user can choose one of the following relations {= - equivalence, ⊆ - superclass, ⊇ - subclass} only after the application of the 1:N matcher. Manual interventions include the following possible actions:

– Adding matched classes' pair,
– Replacing the class belonging to the matched pair with another class,
– Changing the obtained similarity value for a pair of the matched classes,
– Removing the classes' pair that is not matched correctly,
– Editing the threshold value.

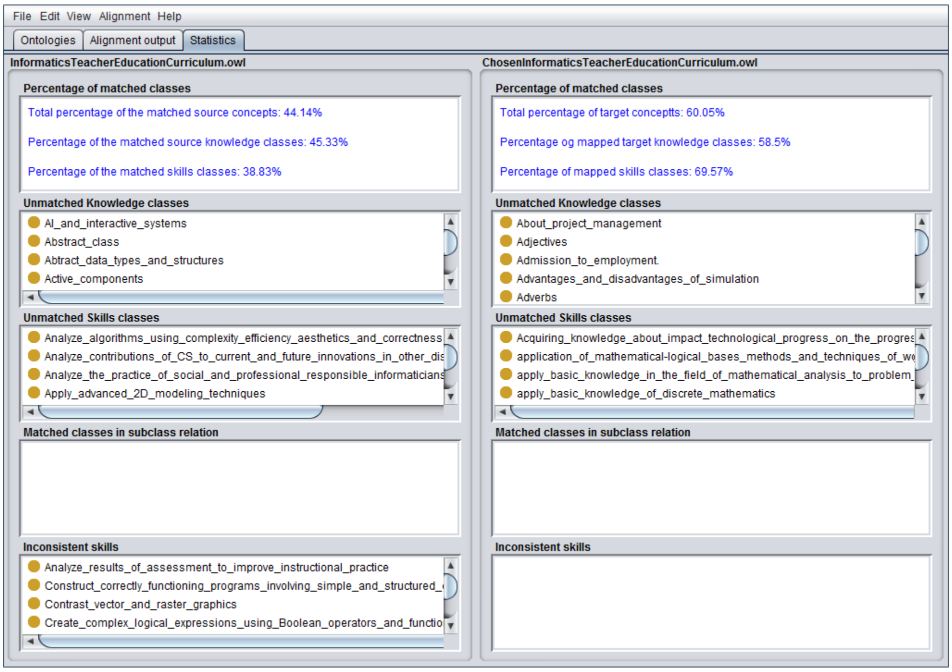

Fig. 4. Statistical presentation of results.

However, adding a matched pair of classes may include:

– Adding a new pair $\{C_{i1}, C_{j2}\}$, where both classes are unmatched.
– Adding a new classes' pair $\{C_{i1}, C_{j2}\}$, where C_{i1} or C_{j2} has already been matched, i.e.: $\exists\{C_{m1}, C_{j2}\} \mid m \neq i \vee \exists\{C_{i1}, C_{k2}\} \mid k \neq j$.
– Adding a new pair $\{C_{i1}, C_{j2}\}$, where both classes are matched, i.e.: $\exists\{C_{i1}, C_{m2}\} \mid m \neq j \wedge \exists\{C_{k1}, C_{j2}\} \mid k \neq i$.

The type of a new relation and the type of the existing relations (for cases 2 and 3) with a class from another ontology are taken into account when creating new pairs of classes. Thus, for example, for the case 2:

– if the user defines the relation of equivalence $\{C_{i1}, C_{j2}\}$, where $\exists\{C_{k1}, C_{j2}\} \mid k \neq i$ then
– the relation $\{C_{k1}, C_{j2}\}$ is deleted and a new pair $\{C_{i1}, C_{j2}\}$ is established, since one class can participate in no more than one equivalence relation with a class of other ontology.

5. Application of the Software Platform to the Ontological Models of the Teacher Education Curricula

Figures 5 -7 show a part of the results obtained after applying the developed software system on the input informatics teacher education models. The column "Source class" contains reference teacher education model' classes, while the classes belonging to the chosen Informatics teacher education model are contained in the "Target class" column.

5.1. The Application of the Algorithm for Calculating the Taxonomic Structural Similarity

Figure 5 shows paired classes and similarity values obtained after applying all three phases of the taxonomic structural algorithm. The possible lower similarity values obtained when comparing the parent classes were taken into account in the experimental determination of the threshold value (70%). Thus, pairs of parent classes (shown in rows 11, 12, 14, 16, 17, 21, 23, 26, 31; Figure 5) have a similarity value below 85%, although they have a similar or identical local name. Such results can be considered as an expected consequence of calculating the similarity of the parent classes which includes the similarities of all subclasses and superclasses. There are also pairs of parent classes whose similarity value is higher, since they belong to very close hierarchical structures (rows 7, 8, 15, 24, 30, 33). It can be seen from Figure 5 that the classes related to the mathematical fields are matched with each other, although only an insight into the names of the classes does not indicate these results (rows 3, 28, 29). However, by looking at the hierarchical structure, it can be concluded that the matching is correct.

When comparing leaf classes, the similarity value more significantly corresponds to the probability that the classes are correctly matched. This is indicated by the pairs of leaf classes shown in rows 6, 9, 13, 19, 20, 25, 32. However, correctly obtained pairs of leaf classes with slightly lower similarity values are possible (rows 2, 10). The pairs of classes contained in rows 5, 18 and 27 can be considered as incorrect results of the application of the taxonomic structural matcher. From Figure 4 it can be seen that the pairs of classes in rows 5 and 18 belong to paired class structures {*Multimedia*, *Multimedia_systems*} and {*Human_Computer_Interaction*, *Interaction_human_computer*}, respectively. These results are a consequence of the principle according to which the system searches, lexically, the closest pairs of classes (with a threshold of 70%) among the subclasses of paired parents. Figure 5 also shows pairs of classes (rows 1 and 4) obtained by applying the third phase of the taxonomic structural matcher, i.e. by searching the classes in unpaired and non-disjoint class structures. The influence of the classes' label on the similarity value can be seen from the correctly obtained pair of classes given in row 22. After applying the taxonomic structural matcher, the percentage of the paired *Knowledge* subclasses of the reference model is 46.1%, while the percentage of pairing of the *Knowledge* subclasses of the selected model is 59.49%.

Row	Source class	Target class	Type of ...	Similarity v...
1	Aggregation	Aggregations	Equival...	100.0%
2	Aims_and_objectives_of_upbringing	Goals_of_education_and_teaching	Equival...	81.18%
3	Algebra	Mathematics_3	Equival...	84.3%
4	Animation	Animation	Equival...	100.0%
5	Audio_formats	Electronic_sound	Equival...	78.33%
6	Complex_numbers	Complex_numbers	Equival...	100.0%
7	Computer_Networks	Computer_networks	Equival...	93.48%
8	Computer_design	Computer_systems	Equival...	87.49%
9	Connection_to_Database	Connecting_to_databases	Equival...	98.0%
10	Criteria_for_quality_of_software	Quality_assessment	Equival...	80.08%
11	Data_structures_and_algorithms	Data_structures	Equival...	83.9%
12	Database	Databases	Equival...	84.92%
13	Deterministic_finite_state_machine	Finite_automata	Equival...	98.06%
14	Didactics	Didactics	Equival...	84.32%
15	E-learning_types	E-learning	Equival...	91.08%
16	Educational_psychology	Pedagogical_psychology	Equival...	75.71%
17	Educational_software	Educational_software_design	Equival...	75.28%
18	Entering_data	Basic_data_type	Equival...	81.67%
19	Entity	Entity	Equival...	100.0%
20	Ethernet	Ethernet	Equival...	100.0%
21	Evaluation_of_instruction	Evaluation_of_teaching_work	Equival...	71.03%
22	Firewalls	Attack_and_protection_of_computer_systems	Equival...	100.0%
23	General_Pedagogy	Pedagogy	Equival...	82.99%
24	Graphics_design_elements	Design_elements	Equival...	98.64%
25	Homomorphisms	Homomorphisms	Equival...	100.0%
26	Human_Computer_Interaction	Interaction_human_computer	Equival...	81.91%
27	Human_diversity_in_HCI	Characteristics_of_man_and_computer	Equival...	71.34%
28	Mathematical_Analysis	Mathematics_2	Equival...	81.11%
29	Mathematics	Mathematics_1	Equival...	89.04%
30	Modeling_and_simulation	Modeling_and_simulations_in_teaching	Equival...	89.04%
31	Models_and_phases_of_the_software_development_pr...	Software_processes_and_specifications	Equival...	74.96%
32	Multimedia_compression_and_decompression	Compression_and_decompression_methods	Equival...	100.0%
33	Programming_Fundamentals	Programming_languages	Equival...	90.56%
34	Software_engineering	Software_engineering	Equival...	85.51%
35	Structured_and_Imperative_programming	Structured_programming	Equival...	90.47%

Fig. 5. Part of the matched parent classes of the teacher education curricula.

5.2. The Application of the Algorithm for Calculating the Relational Similarity

The names of the *Skills* subclass represent the free text contained in learning outcomes. Therefore, the threshold value was experimentally set to a lower value than in the previous phase (55%). The "Bloom" column contains the T mark if the *Skills* subclass of the selected model corresponds to a higher level of the Bloom taxonomy cognitive domain than the corresponding subclass of the reference model. Conversely, the "Bloom" column is false.

R...	Source class	Target class	Type ...	Simil...	Bloom
1	Apply_educational_software	Evaluate_educational_software	Equiv...	85.0%	⊤
2	Contrast_vector_and_raster_graphics	Master_the_basic_concepts_of_computer_graphics	Equiv...	71.6%	⊥
3	Create_e_learning_content	Analysis_of_tools_for_creating_e-learning_systems	Equiv...	72.01%	⊥
4	Design_application_communicationg_and_maintaining_databases	Understand_the_components_of_database_management_software	Equiv...	72.18%	⊥
5	Design_databases	Design_a_database	Equiv...	100.0%	⊤
6	Design_interactive_user_interfaces_for_diverse_applications	Design_user_interfaces	Equiv...	74.7%	⊤
7	Design_web_pages	Creates_a_website	Equiv...	72.32%	⊤
8	Identify_the_national_high_school_CS_curriculum_intending_to_teach.	Knows_the_valid_curriculum_of_informatics_in_primary_and_seco...	Equiv...	62.77%	⊤
9	Implement_basic_algorithms	Problem_solving_through_algorithms	Equiv...	71.29%	⊤
10	Implement_of_programs_of_sufficient_complexity	Implementation_user_interfaces_of_computer_systems	Equiv...	71.52%	⊤
11	Maintain_computer_system	Know_organization_of_computer_systems	Equiv...	70.24%	⊥
12	Select_appropriate_e_learning_approach_to_teach_a_specific_content	Analyze_different_e_learning_approaches	Equiv...	63.61%	⊤
13	Set_the_multimedia_on_the_web	Create_multimedia_presentations	Equiv...	62.74%	⊤
14	Teach_CS_lessons_using_multiple_forms_of_media	Organize_teaching_material_in_the_form_of_educational_software	Equiv...	64.59%	⊤
15	Uderstand_3D_modelling	Create_a_3d_scene	Equiv...	76.89%	⊤
16	Understand_assembler_programming	Use_of_assembly_language	Equiv...	75.19%	⊤
17	Understand_computer_networks_supporting_communication_and_collab...	Configuring_computer_networks	Equiv...	59.2%	⊤
18	Understand_concepts_and_assertions_of_mathematical_analysis_and_...	Acquire_basic_concepts_of_mathematical_analysis	Equiv...	74.13%	⊤
19	Understand_data_representation_and_organization_at_the_machine_level	Understanding_the_work_of_computer_systems	Equiv...	79.3%	⊤
20	Understand_machine_level_components_and_related_issues_of_compl...	Understand_the_structural_organization_of_computers_at_multiple...	Equiv...	67.88%	⊤
21	Use_Modeling_and_simulation_to_solve_real_world_problems	Uses_modelling_and_simulations_in_teaching_informatics	Equiv...	72.28%	⊤
22	Use_UML_for_modeling_meaningful	Application_of_UML	Equiv...	63.05%	⊤
23	Use_of_Internet_in_a_safe_and_efficient_manner	Using_e-mail_services_and_www	Equiv...	57.25%	⊤
24	Use_programs_for_computer_graphics	Uses_raster_graphics_programs	Equiv...	90.63%	⊤
25	Use_programs_for_text_presentations_spreadsheets	Use_of_MS_OFFICE	Equiv...	83.33%	⊤

Fig. 6. Matched *Skills* subclasses of the curricula models.

Considering that the names of the *Skills* subclasses represent a free text and that the similarity value is performed using a terminological matcher (if some of the *Knowledge* classes' structures with which the *Skills* subclasses are associated are matched), matching accuracy is lower as expected. Thus the pairing results shown in rows 2, 3, 10 and 19 can be considered incorrect. The obtained classes' pairs, shown in rows 1, 4, 12, 13, 15, 16, 17 and 18 (Figure 6), regardless of the fact that they represent different levels of the Revised Bloom taxonomy, can be considered as the correct result of matching.

69.57% of the *Skills* subclasses of the selected model are matched after the application of the relational matcher, while the percentage of the matched *Skills* subclasses of the reference model is 38.83%.

The unmatching of the *Skills* subclass is mainly a consequence of the unmatching of the *Knowledge* subclasses with which they are associated. Thus, classes like *Implement_knowledge_representation_and_reasoning_system*, *Assess_possible_ applications_and_limitations_of_the_Artificial_Intelligence* (associated with the unmatched parental class *Artificial_intelligence*), *Discuss_intellectual_property*, *Analyze_the_practice_of_social_and_professional_responsible_informaticians* (associated with the unmatched parental class Computer_ethics), etc. remain unmatched.

5.3. The Application of the Algorithm for Calculating 1:N Similarity

Figure 7 shows characteristic pairs of classes in a 1:N relation. Figure 7 shows an example of a superclass relation. The *Management_in_education* class of the reference curriculum model (described additionally by the "School management" label) has no further subclasses. It is paired with the *Organization_school_work* class of the selected curriculum model, and has become a superclass of all the *Organization_school_work* class' subclasses.

...	Source class	Target class	Type of relati...	Similarity val...
1	Management_in_education	Organization_of_school_work	Equivalence	75.31%
2	Management_in_education	Admission_to_employment	Superclass	75.31%
3	Management_in_education	Development_plan_institutions	Superclass	75.31%
4	Management_in_education	Information_system_primary_and_secon...	Superclass	75.31%
5	Management_in_education	Inspection	Superclass	75.31%
6	Management_in_education	Institution_bodies	Superclass	75.31%
7	Management_in_education	Professional_bodies	Superclass	75.31%
8	Management_in_education	Institutions_for_development_and_qualit...	Superclass	75.31%

Fig. 7. Matched class in 1:N relations.

5.4. Discussion of the Results

After the application of all phases of ontological pairing, there are *Knowledge* and *Skills* classes that remain unmatched. The reasons for their unmatching can be classified into two basic categories. One is the lack of topics (thematic areas) and/or learning outcomes in the compared curricula. The second refers to the possible shortcomings of the applied algorithms, which results in some classes corresponding to the equivalent knowledge/outcomes not being matched. From this it can be concluded that it is either necessary to improve the curricula so that they contain all the required knowledge and outcomes or it is necessary to improve the software platform so that it finds all pairs of classes that represent the same aspects of the compared curricula.

Moreover, a part of the *Knowledge* subclasses remains unpaired as a consequence of the different levels of the description of certain thematic areas in the compared curricula. This especially refers to the pedagogical, didactic and mathematical courses of the selected teacher's curriculum from the Republic of Serbia, which contains a large number of topics (thematic areas). The unmatching of the classes representing these courses does not necessarily mean that they are not included in the compared (reference) curriculum, but may indicate that the same aspects of the curriculum are described by a different number of topics. The possible solution to this type of unmatching is twofold. One direction would be to improve the curriculum by describing the courses in more detail in accordance with the compared curriculum. Another solution is to upgrade the software platform so that algorithms that include “more to more” connections are implemented.

The evaluation of the applied algorithms in our software platform was realized by comparison with test/reference results. Precision and recall are "the most common comparison criteria" [26]. These measures are based on a comparison of the expected and obtained results of the analyzed system. In the context of ontology matching, the alignment obtained from a system that is the subject of evaluation (A) is compared with the reference alignment (R). Precision P is the ratio of the number of correctly found correspondence and the total number of obtained correspondence. Recall R is the ratio of the number of correctly found correspondence and the total number of expected correspondence. It is stated in literature [26] that it is sometimes desirable to consider only one value as a result of comparing the system. However, the systems are often not "comparable" applying only precision or only recall. For example, a system having high recall may have a low precision and vice versa. Therefore, evaluation of the system for ontology alignment [35] usually entails the use of the F-measure that combines precision and recall.

Analogous to [16][17], a team made up of educational experts evaluated the software platform. The expert team consisted of 4 university teachers (in the field of methods of teaching informatics), 2 employees in School Administration (Ministry of Education, Science and Technological Development) and 2 secondary school informatics teachers. The expert team determined the reference alignment (expected pairs of classes) for all possible curricula pairs, i.e. assessed the accuracy of the results obtained by applying the software platform. The process of defining a reference alignment consisted of two main steps. The first was a detailed analysis of the compared ontological models of the curricula done by the team of experts. After that, the expert team defined the reference alignment by finding pairs of classes that represent equivalent knowledge or skills (learning outcomes). In this case, with the exception of the superset/subset relation, one class can be in only one classes’ pair. Thus, the resulting reference alignment contains the exact set of classes‘ pairs that the software platform should ideally provide, according to the expert team. Also, the team of experts analyzed the obtained pairs of classes after applying the platform to the input ontological models. The aim was to determine the number of the correctly obtained pairs of classes as well as the total number of obtained classes, i.e. to determine the values of the parameters needed for calculating precision, recall and f-measure.

In this section the results obtained by comparing teacher education curricula are analyzed regarding the results obtained by comparing the secondary ACM K12 curriculum model and the teacher education reference curriculum model [16], and regarding the results obtained by comparing secondary ACM K12 curriculum model and the chosen teacher education curriculum model [24] in the manner described in [17]. The values of precision, recall and F-measure for all three combinations of input ontological models are shown in Figure 8. Figure 9 shows the percentage of the matched *Knowledge* and *Skills* subclasses.

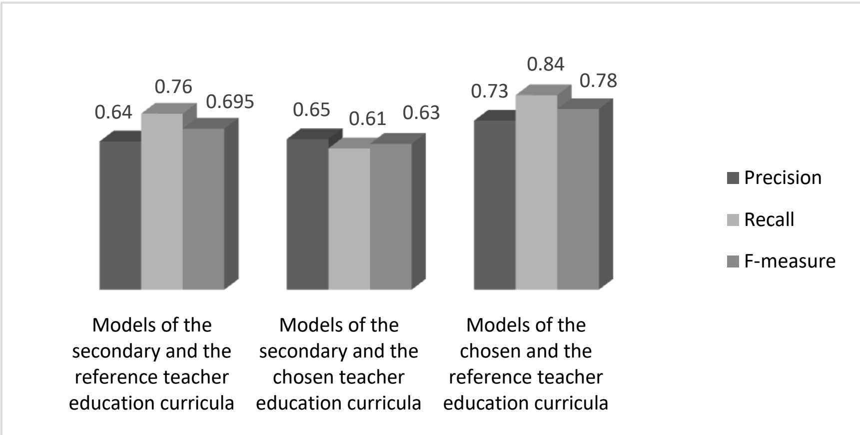

Fig. 8. System evaluation for all three combinations of the ontological models

The analysis shows that the values of precision (0.73), recall (0.84) and F-measure (0.78) are the highest when models of teacher education curricula are compared. Although this can be seen as an expected consequence of comparing the curricula of the same level of education, the results are satisfactory, especially as the largest number of classes was compared (ontological models of teacher education curricula individually contain significantly more classes than high school curriculum model).

It can be noticed that in the first and third case (Figure 8) the higher values of recall than the values of the precision were obtained (in the second case the value of recall is close to the value of precision). These results (along with satisfactorily high precision value) are in accordance to the reference [36] where "highest priority" is given to the recall when the ontological matching is a semi-automatic process [16]. In [36], (p 630) states that “since the burden of deleting false identified pairs by a platform is minimal compared to the burden of traversing two heterogeneous ontologies that might include thousands of concepts and attributes and identify similar entities, recall is a much more important measure”.

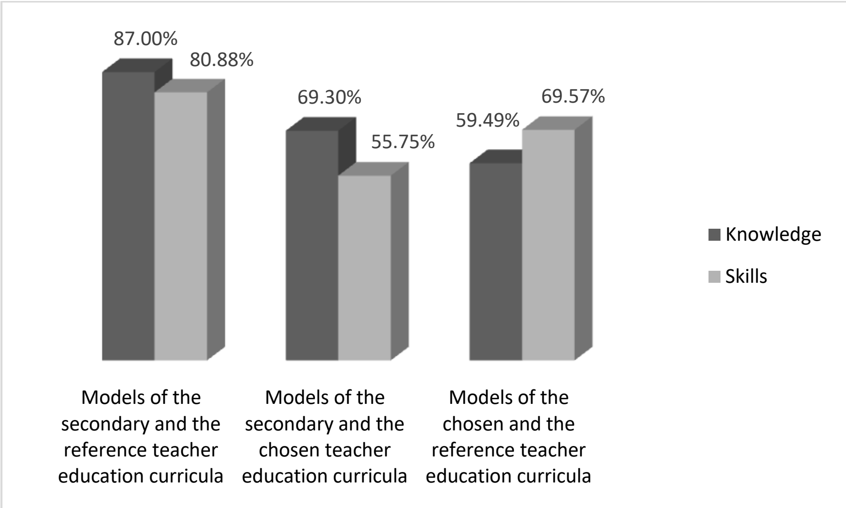

Fig. 9. The percentage of matched classes for all three combinations of the ontological models

From Figure 9 it can be observed that the percentage of the matched *Knowledge* subclasses of the secondary school model is lower when it is compared with the chosen teacher education curriculum model than when it is compared with the reference teacher education curriculum model. This could be explained by different taxonomical structure of the ontological curricula models (structurness of the chosen teacher education curriculum model is the lowest). That is especially true for "Connection between mathematics and computer science" topic [16], [17]. Still, a lower percentage of matching in the second case (Figure 9) primarily indicates that the chosen teacher education curriculum does not provide the study of thematic areas corresponding to the unmatched classes of the secondary model.

The higher level of matching between the secondary model and the reference teacher education curriculum model is expected, since the reference model of the teacher education curriculum was created according to the recommendations of international accreditation bodies and the analysis of more than 20 national and international curricula.

The percentages of the matched *Knowledge* subclasses are the lowest when teacher education curricula models are compared (third case in Figure 9). For such combination of input ontological models, a higher percentage of the matched *Skills* subclasses is obtained (compared to matched *Knowledge* subclasses). These results can be considered as a consequence of a large number of differently structured *Knowledge* subclasses when compared teacher education curricula models. Still, a large number of *Knowledge* subclasses is "correctly unmatched" (there are missing thematic areas/courses in the compared curricula). The percentages of the matched *Knowledge* and *Skills* subclasses of the reference model (the opposite case) are not shown in the table and are similar to the results of the matching of the chosen teacher education curriculum model. The results of the comparison of teacher education curricula models (section 5) are in accordance with the results of the comparison of the other combination of input ontological models. For example, when comparing the selected teacher education curriculum model and ACM K12 model, the system has shown that the concepts of artificial intelligence are not taught in the selected teacher education curriculum in the Republic of Serbia. On the other hand, when comparing the ACM K12 model and the reference model of the teacher education curriculum [17], classes' structures that correspond to these concepts are mutually matched. From these results it can be assumed that the correct result of comparing teacher education curricula models would be an unmatched class of the reference model representing the principles of artificial intelligence, which is obtained by the application of the software platform.

6. Conclusions and Future Work

The semi-automatic software platform presented in this paper contains modified ontological matching algorithms, which enables the comparison of ontological curricula models of the same level of education i.e. informatics teacher education curricula. The main contributions of this paper are threefold: 1) model of the selected informatics teacher education curriculum has been created 2) the part of matching algorithms, adapted to teacher education curricula models, has been developed 3) comprehensive evaluation of the software platform and curricula models has been conducted. The evaluation was realized by unifying and comparing the results obtained for all three combinations of input ontological models. The results indicate the lack of specific knowledge (representing pedagogical, didactic and mathematical thematic areas) and skills in the analyzed curricula, but also the different structure of the ontological models. Therefore, it is necessary to consider the improvement of the curricula as well as the introduction of new matching algorithms that would find equivalent class belonging to related hierarchical structures. Also, the values of precision, recall and f-measure are lower when comparing curricula of different levels of education than when comparing teacher education curricula. Hence, in the case of matching secondary and teacher education models, the need for manual user interventions is greater, which can be considered as the expected result of the software platform application. Inversely, the matching of the *Knowledge* and *Skills* subclasses is greatest when comparing the secondary and teacher education curriculum (especially the reference model of teacher education curriculum).

One of the directions of further research refers to the creation of an ontological model in such a way that it also contains other important aspects of the curriculum, such as assessment methods, learning goals, anticipated literature and the like. Also, it is necessary to explore the possibility of a semi-automatic mapping of informatics curricula to ontological models. Other directions of future research are related to the limitations of the software platform and the possibilities of improving its accuracy. This primarily refers to the applied matching algorithms. Thus, when comparing a series of words, either in the name of thematic areas (*Knowledge* subclasses) or in the name of learning outcomes (*Skills* subclasses), the terminological matcher does not take into account the nature of the domain (Computer science/Informatics) or the syntax of the language. The similarity of the classes' name depends on the similarity of the separate words (tokens). For example, in the WordNet dictionary, the same word for computer science domain can have a completely different meaning. Thus, the word ontology (WordNet Search, version 3.1) is defined as "a rigorous and exhaustive organization of some knowledge domain that is usually hierarchical and contains all the relevant entities and their relations" (for Computer science domain) or "the metaphysical study of the nature of being and existence" (in general). Therefore, it is necessary to consider improvements in the terminological matcher in some of the following ways: using the semantic domain of WordNet dictionaries, using external dictionaries, using external computer ontologies or using ACM computer classification.

Since English can be seen as a de facto standard for international recommendations for Computer science curricula (ACM, CSTA) the software platform uses the English version of the WordNet dictionary and the ontological models are written in English. Therefore, another future research aim is to provide conditions for comparing ontological models in different languages (using the "Inter-Lingual Index" component of the WordNet dictionary) or comparing models whose classes' names are in the same language.

Also, future work will be focused on improving the system performance by applying the method for the initial rejection of classes that will not be considered. Also, it is necessary to investigate the accuracy of results with manual interventions of users in different stages of alignment. One more research aim is to map more curricula onto ontological models and to use the software platform to examine the accuracy of the results and the compliance of the curricula.

References

1. Seitz, P.: Curriculum Alignment Among the Intended, Enacted, and Assessed Curricula for Grade 9 Mathematics. Journal of the Canadian Association for Curriculum Studies, Vol. 15, No. 1, 72- 94. (2017).
2. Bay E.: Developing a Scale on Factors Regarding Curriculum Alignment. Journal of Education and Training Studies. Vol. 4, No. 5, 8-17. (2016)
3. Wijngaards-de Meij, L., Merx, S.: Improving curriculum alignment and achieving learning goals by making the curriculum visible. International Journal for Academic Development, Vol. 23, No. 3, 219-231. (2018)
4. Shaltry, C.: A new model for organizing curriculum alignment initiatives. Advances in physiology education, Vol. 44, No. 4, 658 – 663. (2020)
5. Ruge, G., Tokede, O., Tivendale, L.: Implementing constructive alignment in higher education – cross-institutional perspectives from Australia. Higher Education Research & Development, Vol. 38, No. 4, 833-848. (2019)
6. Zhu, Y C., Zhang, W., He, Y., Wen, J B., Li, M Y.: Design and Implementation of Curriculum Knowledge Ontology-Driven SPOC Flipped Classroom Teaching Model. Educational Sciences: Theory & Practice, Vol. 18, No. 5, 1351-1374. (2018)
7. Katis, E., Kondylakis, H., Agathangelos, G., Vassilakis, K.: Developing an Ontology for Curriculum and Syllabus. In: Gangemi A. et al. (eds) The Semantic Web: ESWC 2018 Satellite Events. ESWC 2018. Lecture Notes in Computer Science, Vol. 11155, 55-59. (2018)
8. Elsayed, E.: Interaction with Content through the Curriculum Lifecycle. Advanced Learning Technologies, ICALT 2009, 730 – 731. (2009)
9. Sarmiento, C., Duarte, O., Barrera, M., Soto, R: Semi-automated academic tutor for the selection of learning paths in a curriculum: An ontology based approach. Proc. of IEEE 8th International Conference on Engineering Education, 223-228. (2016)
10. Zhu, Y. C., Zhang, W., He, Y., Wen, J. B. , Li, M. Y.: Design and implementation of curriculum knowledge ontology-driven SPOC flipped classroom teaching model. Educational Sciences: Theory & Practice, Vol. 18, No. 5, 1351–1374. (2018)

11. Modiba, N., Ojo, S., Ncube Z.: An Ontology Based Model for Cyber Security Awareness Education, in Kennedy Njenga (editor). Proceedings of 4th International Conference on the Internet, Cyber Security and Information Systems, Vol 12, 169–179. (2019)
12. Ngo, D., Bellahsene, Z.: Overview of YAM++—(not) Yet Another Matcher for ontology alignment task. Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 41, 30–49. doi:10.1016/j.websem.2016.09.002 (2016)
13. Fiallos, A., Ochoa, X.: Semi-automatic generation of intelligent curricula to facilitate learning analytics. Proceedings of the Ninth International Conference on Learning Analytics & Knowledge, Tempe, AZ, USA, 46–50. (2019).
14. Shao, C., Hu, LM., Li, JZ.et al.: RiMOM-IM: A Novel Iterative Framework for Instance Matching. Journal of Computer Science and Technology, Vol. 31**,** 185–197. (2016). https://doi.org/10.1007/s11390-016-1620-z
15. Nkisi-Orji, I., Wiratunga, N., Massie, S., Hui, KY., Heaven, R.: Ontology Alignment Based on Word Embedding and Random Forest Classification. Springer Theses, 557–572. (2019) doi:10.1007/978-3-030-10925-7_34
16. Mandić, M., Konjović, Z., Ivanović, M.: Platform for Computer-aided Harmonization of Informatics Curricula. Acta Polytechnica Hungarica, Vol. 13, No. 3. 159-179. (2016)
17. Mandić, M., Konjović, Z.: Collaborative development of informatics curricula based on semantic technologies. International Scientific Conference On Ict And E-Business Related Research – Sinteza, Ict & Business, 3-9. (2016)
18. Mandić, M., Konjović, Z., Ivanović, M.: Ontological Model of the Standardized Secondary School Curriculum in Informatics. Proceedings of the 5th International Conference on Information Society and Technology, 363-367. (2015)
19. Stephenson, C., Gal-Ezer, J., Haberman, B., Verno, A.: The new educational imperative: Improving high school computer science education. Final report of the CSTA Curriculum Improvement Task Force. New York: Computer Science Teachers Association. (2005)
20. Fuller, U., Johnson, CG., Ahoniemi, T., Cukierman, D., Hernán-Losada, I. et al.: Developing a computer science-specific learning taxonomy. ACM SIGCSE Bulletin, Vol. 39, No. 4, 152-170. (2007)
21. Gal-Ezer, J., Stephenson, C.: Computer science teacher preparation is critical. ACM Inroads. Vol. 1, No. 1, 61-66. (2010)
22. East, JP., Bentley, C., Kmoch, J., Rainwater, S., Stephenson, C.: NCATE standards for preparation of secondary computer science teachers. Proceedings of the 42nd ACM technical symposium on Computer science education, 243-244. (2011)
23. Hazzan, O., Ragonis, N., Lapidot, T.: Guide to Teaching Computer Science.Springer, Cham. (2020). doi:10.1007/978-3-030-39360-1
24. Technical faculty "Mihajlo Pupin": Informatics and techniques in education, Zrenjanin. (2013). [Online].http://www.tfzr.uns.ac.rs/Content/akreditacija/Informatika%20i%20tehnika%20u%20obrazovanju%20master.pdf
25. Ontological models of the informatics teacher education curricula. http://www.pef.uns.ac.rs/InformaticsTeacherEducationCurriculum/index.html
26. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer-Verlag, Berlin-Heidelberg. (2007)
27. Wu, W., Yu, C., Doan, A., Meng, W.: An Interactive Clustering-based Approach to Integrating Source Query interfaces on the Deep Web. Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, 95-106. (2004)
28. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity Flooding: A Versatile Graph Matching Algorithm. In *Proceedings of the 18th International Conference on Data Engineering*. (ICDE), 117-128. (2002)
29. Huber, J., Sztyler, T., Noessner, J., Meilicke, C.: CODI: Combinatorial Optimization for Data Integration : results for OAEI 2011. In Proc. 6th ISWC workshop on Ontology Matching (OM), 134–141. (2011)

30. Ngo D, Bellahsene Z, Coletta R.: YAM++ - Results for OAEI 2011. In Proceedings of the 6th International Workshop on Ontology Matching (OM-2011), Vol. 814, 228–235 (2011)
31. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, IF., Couto, FM.: The agreementmakerlight ontology matching system. In: Meersman, R., Panetto, H., Dillon, T., Eder, J., Bellahsene, Z., Ritter, N., De Leenheer, P., Dou, D. (eds.); Heidelberg: Springer; LNCS, vol. 8185, 527–541. (2013)
32. Lin, D.: An information-theoretic definition of similarity. *Proceedings of the 15th International Conf. on Machine Learning*, 296–304. (1998)
33. Jaro, M.: UNIMATCH: A record linkage system: User's manual. Technical report. Washington DC: U.S. Bureau of the Census. (1976)
34. Jaro, M.: Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. Journal of the American Statistical Association, Vol. 84, No. 406, 414–420. (1989)
35. Grau, BC., Dragisic, Z., Eckert, K., Euzenat, J, et al.: Results of the ontology alignment evaluation initiative 2013. In: *Proc. 8th ISWC workshop on ontology matching (OM)*. 61–100 (2013)
36. Stoilos, G., Stamou, G., Kollias, S.: A String Metric for Ontology Alignment. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) *ISWC* 2005. LNCS, Vol. 3729, 623–637. (2005)

Milinko Mandić received his MSc degree in the field of electrical and computer engineering from the Faculty of Technical Sciences, University of Novi Sad, Serbia. He obtained his PhD degree in Methods of teaching informatics from the Faculty of Sciences, University of Novi Sad. He is currently an Assistant Professor at the Department of Informatics and Media, Faculty of Education, University of Novi Sad. His research interests include informatics curricula (for informatics teachers and for the primary and secondary level of education), Semantic Web and ontologies, methods of teaching informatics, e-learning standards, applying social and collaborative software in education.

Received: February 07, 2021; Accepted: July 08, 2021.