INFORMATION & COMMUNICATION TECHNOLOGY

COLLABORATIVE DEVELOPMENT OF INFORMATICS CURRICULA BASED ON SEMANTIC TECHNOLOGIES

Milinko Mandić[1],
Zora Konjović[2]

[1] University of Novi Sad,
Faculty of Education,
Sombor, Serbia

[2] Singidunum University,
32 Danijelova Street, Belgrade, Serbia

Abstract:

In order to keep pace with the ongoing changes in ICT and increasing common IT competencies requirements, informatics curricula at secondary school level and, consequently, curricula educating informatics teachers must be frequently changed to ensure necessary competencies. This paper proposes collaborative development of informatics curricula assisted by a software tool for compatibility analysis of secondary school informatics curricula and curricula by which teachers of informatics are educated. The proposed software tool relies upon semantic technologies, *i.e.* ontologies for representation of competence-based curricula and ontology alignment for compatibility analysis. The secondary school informatics curriculum ontology was built to comply with the ACM K12 standard, while the teachers' curriculum ontology was built based on the selected existing curricula. The paper presents a brief description of the software tool and the results of the domain (informatics) segment of teachers' curriculum offered by the selected Serbian university and the standardized ACM K12 compliant secondary school informatics curriculum.

Key words:

education, informatics, curriculum, ontology, ontology alignment.

1. INTRODUCTION

The intensity and speed of change in the ICT field require continuous improvement of the informatics curricula, which calls for curriculum representation facilitating its simple and fast creation, modification and maintenance. Curricula creators, authorized ministries, accreditation bodies, teachers, *etc.* should also have THE opportunity to collaborate in curricula compliance monitoring and verification.

The current literature (Chi, 2009; Demartini *et al.*, 2009; Elsayed, 2009; Fernández-Breis *et al.*, 2012) indicate that ontologies and Semantic Web technologies are increasingly used for curriculum representation.

Ontology matching is the process of finding correspondences between entities of compared ontologies. Ontology alignment is a set of correspondences between two or more ontologies (Euzenat and Shvaiko, 2007) and represents the output of the matching process.

There are numerous techniques and strategies for ontology matching applied to different systems such as (Li *et al.*, 2009; Cruz *et al.*, 2007; Jean-Mary *et al.*, 2009). However, to our best knowledge, only (Mandić, Konjović, and Ivanović, 2016) provides the implementation of the system

Correspondence:

Milinko Mandić

e-mail:
milinko.mandic@pef.uns.ac.rs

that utilizes ontology alignment for determining education curricula matching.

This paper presents the software platform for semi-automatic determination of compliance between informatics teacher education curriculum and secondary informatics curriculum. Since the curricula are firstly represented by ontologies, techniques and algorithms for ontology matching are applied to check compliance of the curricula, whereby the users can change the results of ontology alignment.

The paper comprises five sections and the list of references. The second section is about ontological models of the curricula, the third one briefly describes the architecture of the software tool, the fourth one presents the results of the compatibility analysis obtained by applying the proposed software tool to the domain (informatics) segment of teachers' curriculum offered by the selected Serbian university and standardized ACM K12 compliant secondary school informatics curriculum. The last section concludes on the achieved results and elaborates on further research.

2. ONTOLOGICAL MODELS OF THE CURRICULA

The aim of this research is essentially to provide software tool that should give a hand in determining whether a teacher gains competencies to teach informatics in a high school upon completion of a study programme, as prescribed by the secondary school informatics curriculum. For that reason, models of teachers' and secondary schools' informatics curriculum are based on competencies. The principal class of ontological model of both teacher education and secondary school curriculum is *Competence*. Competence model is presented in detail in (Mandić, Konjović, and Ivanović, 2016; Mandić *et al.*, 2013; Mandić, Konjović, and Ivanović, 2015).

The ontological model of a secondary school informatics curriculum is created on the basis of the ACM K12 CS proposal, which is the standardized curriculum accepted worldwide. This ontology is minutely described in (Mandić, Konjović, and Ivanović, 2015).

In this paper, we choose the teacher education curriculum "Technics and Informatics" offered by the Faculty of Technical Sciences in Čačak for the analysis. This curriculum has been chosen for the following reasons: detailed information about the study program is available on the official website; integrated studies last for 5 years and give a master's degree; studies have been accredited in both of

the accreditation processes carried out in the Republic of Serbia. Additionally, Faculty of Technical Sciences (2013) (Standard 1): "study program has been founded on the tradition in education of teachers of technical disciplines at the Faculty of Technical Sciences in Čačak" and "is a continuation of the study programs for the education of technical education teachers (implemented since 1975), technics and informatics teachers (implemented since 1993), informatics teachers (implemented since 2006)".

Upper classes of ontological representation of the chosen curriculum are modeled in accordance with the ontological representation of competencies based on IEEE RCD standard, described in detail in (Mandić, Konjović, and Ivanović, 2015). Following the reference teacher education curriculum proposed in (Mandić, Konjović, and Ivanović, 2016), class *Knowledge* has 5 direct subclasses (Figure 1).
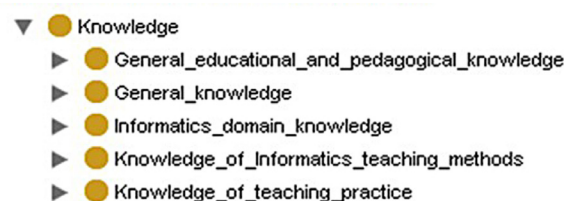

Fig. 1. Direct sublclases of the *Knowledge* class

All courses (compulsory and optional) specified in the Standard 5 of accreditation materials (Faculty of Technical Sciences, 2013) are mapped in the ontological model. The courses are categorized into one of the five general fields: general knowledge, general educational and pedagogical knowledge, informatics domain knowledge, knowledge of teaching practice, and knowledge of informatics teaching methods. The principle of mapping the curriculum courses in the ontological model was as follows. Because of the need that the terms contained in the curriculum are, as much as possible, "machine understandable" and comparable to the terms contained in other created curricula, all the courses' titles, as well as the courses' outcomes and content, were first translated into English. The titles of courses were mapped to the subclasses of classes that represent appropriate general curriculum fields, depending on the domains in which they were classified. Thus, for example, the classes *Physics* and *Probability_and_Statistics* were mapped to subclasses of the *General_knowledge* class; the classes *Pedagogy* and *Psychology* became subclasses of the *General_educational_and_pedagogical_ knowledge* class, *etc.*

The subclasses of the *Informatics_domain_knoweldge* class are shown in Figure 2.

The next step in creating the ontological model was to represent topics described in the course content as subclasses of classes that correspond to courses. If a topic contains several other topics/subtopics (often presented as enumeration), then all the listed subtopics are mapped as its subclasses. Also, if some topic is further described, further clarification is mapped to its label. In the last phase of creating the ontological model, course outcomes are mapped in accordance with (Churches, 2008; Heer, 2012; Krathwohl, 2002) to the appropriate subclasses of the classes represented by Revised Bloom's taxonomy and connected via *hasKnowledge/hasSkills* object properties with the class corresponding to the course they belong to.

Fig. 2. *Informatics_domain_knowledge* class architecture

The ontological model of informatics teacher education curriculum is available at www.pef.uns.ac.rs/ SerbianInformaticsTeacherEducationCurriculum/index.html.

3. SOFTWARE ARCHITECTURE

Software components

Figure 3 shows software architecture of the tool based on Semantic Web technologies, which is aimed to assist curricula synchronization. The software tool was developed in the programming language Java.

The software architecture is based on the architecture of the AgreementMaker system proposed in (Cruz *et al.*, 2007). The basic components of the software are:

- Ontology converter
- Ontology alignment system
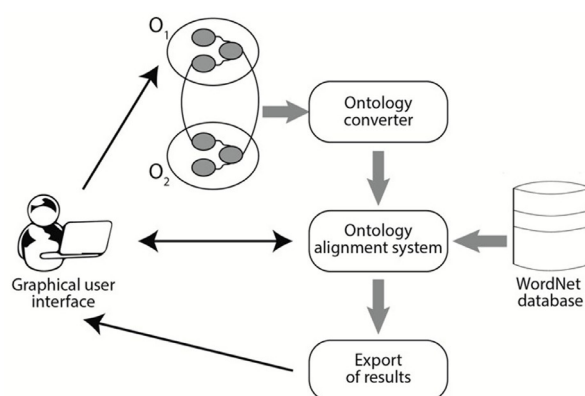- Export of results
- Graphical user interface

Fig. 3. System architecture for curricula synchronization

Ontology converter

This component enables converting OWL/RDF ontological models of curricula into appropriate classes in the Java programming language. Ontology converter is the software tool component that provides reading and manipulating of input ontologies, and therefore is the point where collaboration in development of ontologies can take place.

Ontologies are converted by using Protégé OWL API (http://protegewiki.stanford.edu/wiki/ProtegeOWL_ API_Programmers_Guide) and Jena API (https://jena. apache.org/).

Ontology alignment system

The semi-automatic system provides an alignment of input ontological models by applying ordered sequence of algorithms for calculating firstly the terminological similarity (using WordNet database), then taxonomic structural similarity and finally relational similarity. Possible relations between compared ontologies classes are the following ones: equivalence, superclass, and subclass.

Once the tokenization is performed, terminological similarity is determined by comparing the obtained tokens contained in classes' local names and labels by using Lin (1998) and Jaro-Winkler measures (1989, 1999).

Taxonomic structural similarity of the observed classes takes into account similarities of all superclasses (not just the direct one), similarities of all subclasses as well as previously calculated terminological similarity.

Relational similarity is based on the principle that if two classes that represent the domain of object property (relationship) are similar, and if object property is similar, then the range classes connected with domain classes via that object property are similar.

All algorithms are described in detail in (Mandić, Konjović, and Ivanović, 2016).

Export of results

This component provides tabular and statistical display of the obtained results. It provides the basis for further collaborative (domain experts, professional bodies, ministries, informatics teachers) analysis and curricula improvement. Export of results should enable converting the results into recommended formats (Euzenat and Shvaiko, 2007) for displaying the results of harmonization (such as Alignment format, the OWL *etc.*), which has not been implemented in the prototype software yet.

Graphical user interface

Graphical user interface (GUI) allows users an insight into the structure of ontologies in the side-by-side view, the properties of the classes, and results of the matching. In the next section of the paper, GUI is shown through examples of displaying system's characteristic results (Figures 4, 5 and 6)

4. ANALYSIS OF RESULTS

This section presents the results of aligning the secondary informatics curriculum and the selected teacher education curriculum. Presentation, which contains the characteristic results and their analysis, follows the steps executed after the terminology alignment already took place.

Taxonomic structural similarity

As an example of the system output, part of the matched classes after the application of the first taxonomic structural similarity algorithm is shown in Figure 4.

The columns "Source class" and "Target class" contain classes' local names of the ontological representation of

the secondary school and chosen teaching curriculum, respectively; column "Type of relation" means the type of connection between the classes ("Equivalence", "Superclass" or "Subclass"), while "Similarity value" contains calculated value of similarity between matched classes. The threshold value was set to 70%.

Row	Source class	Target class	Type of relation	Similarity value
1	Algorithms_and_data_structures	Data_types	Equivalence	87.91%
2	Applications	Computer_applications	Equivalence	79.2%
3	Computer_networks	Computer_Networks_and_Co...	Equivalence	87.87%
4	Connections_Between_Mathematics...	Mathematics_of_Informatics	Equivalence	84.14%
5	Data_Structures	Complex_Types	Equivalence	72.96%
6	Databases	Databases	Equivalence	90.08%
7	Fundamentals_of_Hardware_Design	Memory_Systems_and_Techn...	Equivalence	85.19%
8	Internet_elements	Internet	Equivalence	79.44%
9	Multimedia	Multimedia_systems	Equivalence	79.01%
10	Object-oriented_programming	Object-Oriented_Programming	Equivalence	94.22%
11	Operating_Systems	Operating_systems	Equivalence	88.32%
12	Parts_of_a_Computer	Practicum_in_computer_archit...	Equivalence	75.64%
13	Principles_of_computer_organization	Computer_architecture	Equivalence	82.05%
14	Programming_Languages	Programming_Languages	Equivalence	85.15%
15	Programming_and_software_engine...	Introduction_to_programming	Equivalence	73.15%
16	Relational_database_design	Relational_databases	Equivalence	82.23%
17	Representing_Information_Digitally	Information_Technology	Equivalence	70.99%
18	Standard_software	Classification_of_Applications	Equivalence	75.21%
19	Structured_programming	Control_Structures	Equivalence	82.22%
20	The_major_component_parts_of_the...	Languages_Pascal_and_C	Equivalence	72.12%

Fig. 4. Part of the matched parent classes

Here, one can observe that similarity of classes' pairs with the same or very similar names is not necessarily close to or equal to 100%. Typical examples are pairs of classes shown in rows 2, 6, 9, 11, and 14 of the Figure 4. The reason for that is the participation of subclasses and superclasses similarities, if any, in the overall similarity of the observed parental classes. Thus, the pair of classes {*Applications, Computer_applications*} (line 2, Figure 4) has a similarity below 80% due to the differences in structure and subclasses names (Figure 5). For example, database topics in the teacher education curriculum is represented by a detached course, not modeled as a subclass of the *Computer_applications* class.

Figure 4 contains pairs of classes that can be considered inadequate, the typical example is the pair of classes shown in row 20.

On the other hand, classes *Fundamentals _of_Hardware_ Design* and *Memory_Systems_and Technologies*, although with different meanings of local names, have been properly matched (Figure 4, line 7). After examining the hierarchical structure of the compared ontologies (Figure 5), it can be seen that the class *Fundamentals_of_Hardware_Design* includes subclasses corresponding to subclasses of the class *Memory_Systems_and_Technologies.*

Analogous conclusion can be drawn for the pair of classes shown in row 12, Figure 4.

The final percentage of the matched *Knowledge* subclasses obtained at this stage is 62.99%.
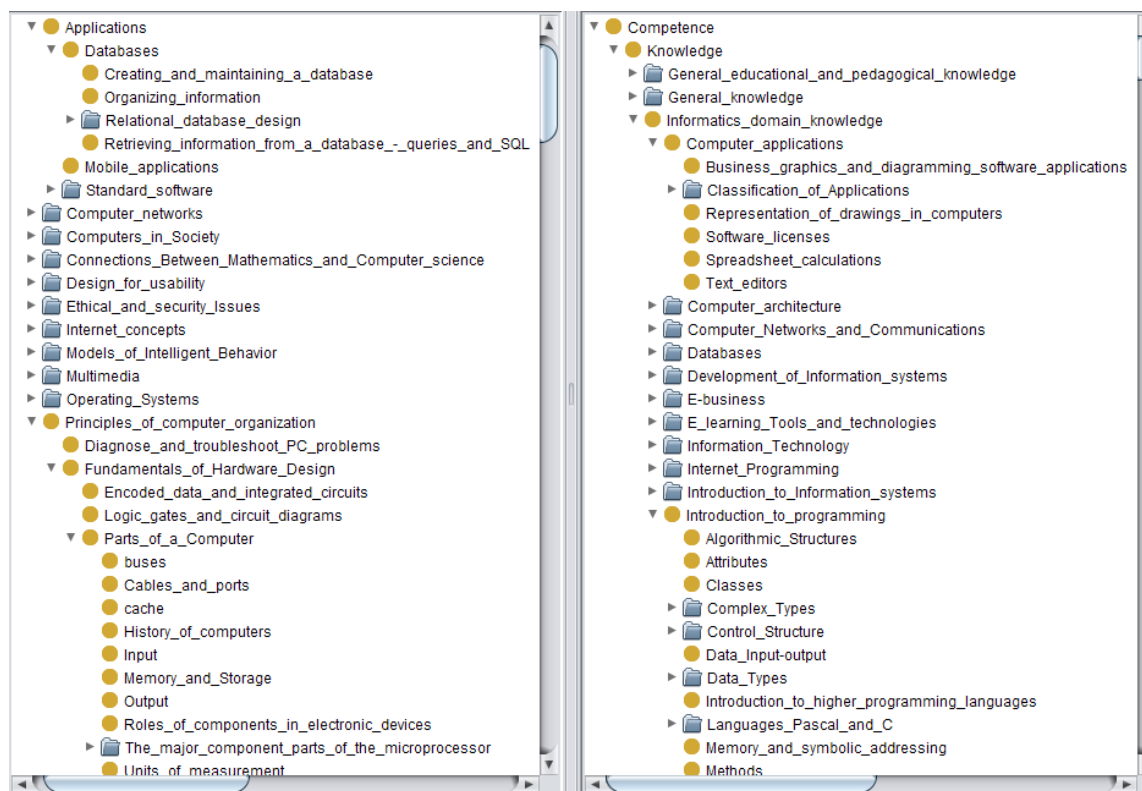
Fig. 5. Fragments of hierarchical structures of the secondary school and teacher curriculum models

Fig. 6. Examples of matched *Skills* subclasses

Relational similarity

Since the *Skills* class has quite poor structure and classes' names and labels in this part of ontology usually contain a large number of words, lower values are obtained for similarities between *Skills* subclasses. Therefore, threshold is here set at lower value (55%). Some of the pairs of *Skills* subclasses obtained by using an algorithm for determining the relational similarity is shown in Figure 6.

The column "Bloom" in Figure 6 contains ⊤ mark if the level of skills in the teaching curriculum is higher (which is desirable) than the required level in the

secondary school curriculum or the mark ⊥ if that is not the case. Figure 6 shows that one can observe that this matching phase also gives classes pairs that can be construed inappropriate. This refers, above all, to pairs of classes shown in rows 2, 3, 12 and 17. Namely, the classes *Connect_a_computer_to_a_network* and *Diagnose_and_fix_ faults_in_the_network* (row 3, Figure 6) are associated with subclasses *Computer_networks* and *Computer_Networks_and_Communications* of the class *Knowledge* respectively, which are matched by algorithm for determining the taxonomic structural similarity (row 3, Figure 4). Therefore, at this stage the appropriate class

The Internet and Development Perspectives

for the source class *Connect_a_computer_to_a_network* is obtained by searching the set of *Skills* subclasses with which the class *Computer_Networks_and_Communications* is connected. However, the corresponding outcomes indicate the need to manually intervene in automatically obtained results.

Examples of classes pairs where class of secondary school model represents the outcome of a higher level of Bloom's taxonomy than the level to which appropriate skills (class) of teaching curriculum belong are shown in rows 11, 13, 19 and 20.

The percentage of matched *Skills* subclasses is 51.66%.

System evaluation and main conclusions

Software platform validation has been carried out by the team of educational experts formed to represent the profiles of potential users of the system. The team is made up of 4 university teachers (elected to the position of informatics teaching methods), 2 employees in School Administration (Ministry of Education, Science and Technological Development) and 2 secondary school teachers of informatics. Their task was to determine the expected class pairs (reference alignment) for input curricula ontological models. On the basis of the reference alignment and the obtained system output, precision, recall and F-measure are determined.

Based on the received percentage of matched *Knowledge* and *Skills* subclasses (with a precision of 0.63, the recall of 0.58 and F-measure of 0.60), it can be concluded that the teacher education curriculum "Technics and Informatics" does not envisage the study of a number of important topics of secondary level ACM K12 standard such as the constraints of computing, the principles of artificial intelligence, ethical and security issues, the principles of creating user friendly interfaces, as well as the deeper study of the principles of problem solving and algorithms.

Part of unmatched classes is the result of different taxonomic structure of secondary model and the analyzed teacher education curriculum. This is especially true for the topics of robotics and connections between mathematics and computer science, which are mapped to the classes that correspond to non-domain knowledge in the chosen teaching curriculum model.

Based on the results of the unpaired *Skills* subclasses, one can conclude that the analyzed teacher education curriculum does not include outcomes such as: describe major applications of artificial intelligence, differentiate between ethical and legal issues, evaluation of algorithms

based on their efficiency, accuracy and clarity, evaluation of usefulness of the user interface, testing program code, *etc.*

Matched *Skills* subclasses of the model of the chosen teaching curriculum that represent inappropriate level of Bloom's taxonomy are usually related to programming skills. They are often described using verbs that do not correspond to the highest level of cognitive dimensions of the Revised Bloom's taxonomy. Examples of such outcomes in teaching curriculum (those mapped on the subclasses of the classes *Apply* or *Remember-understand*) are: "knows about the basic types and data structures", "understands and applies the work with arrays and structures" (the course 'Introduction to Programming'); "knows the concept of object-oriented approach, and uses classes and objects in the programming language C ++" (the course 'Programming Languages'). It is possible to derive two main conclusions based on the described inconsistencies. The first one appertains to the improvement of the ontological model of the curriculum, and the other one to the improvement of the teacher education curriculum aimed at providing competences that better satisfy needs of a secondary school curriculum. Within the context of improving the ontological representation, it is possible to reconsider the manner of representation of certain outcomes so that the verbs that describe the outcome do not represent the key criteria for mapping to subclass of Bloom's taxonomy within the created model. In the case of the outcome "use classes and objects in the programming language C++", it is therefore necessary to decide whether the use of classes and objects in the specific object programming language is the essence of programming (which belongs to the highest level of Bloom's taxonomy). If yes, the specific outcome should then be mapped to the appropriate subclass of the *Create* class. Another conclusion that can be derived from described inconsistency is the improvement of curriculum so that it contains outcomes that, in the field of programming, correspond to the highest level of Bloom's taxonomy.

CONCLUSIONS AND FUTURE WORK

The results obtained for curricula matching and the results of the system evaluation show that the software tool developed and presented in this paper can facilitate harmonization of the standardized secondary school curriculum and teacher education curriculum. As shown by the case study, application provides users with clear and useful information on the knowledge and skills/outcomes missing in the selected teacher education curriculum.

On the other hand, it also provides information useful for better description of the curriculum, *e.g.* outcomes in the field of programming that correspond to the appropriate level of the Revised Bloom's taxonomy. The obtained incorrect matching justifies the need for manual intervention in the alignment results, which particularly refers to the *Skills* subclasses. Better matching in this part of ontologies calls for better standardization in defining curricula learning outcomes.

Although the research subject opens a large space for further work, we mention only two directions here. The first one (more technical) is development of software components for collaboration (*e.g.*, standardized formats for data exchange, *etc.*) enabling users (IT experts, teachers, administrative officials, *etc.*) to take part in collaborative changes of the alignment results. The second one, which is substantially important, is further improvement of curricula models and ontology alignment component (*e.g.*, competence model improvement, introduction of more sophisticated alignment algorithms, *etc.*).

REFERENCES

A. Churches, "Bloom's Digital Taxonomy," 2007, http://www.techlearning.com/techlearning/archives/2008/04/andrewchurches.pdf

A. Elsayed, "Interaction with Content through the Curriculum Lifecycle," Advanced Learning Technologies, ICALT 2009, 2009, pp. 730 – 731.

D. Lin, "An information-theoretic definition of similarity," Proceedings of the 15th International Conf. on Machine Learning, 1998, pp. 296–304.

D. R. Krathwohl, „A revision of bloom's taxonomy: An overview," Theory into Practice, vol. 41, no. 4, 2002, pp. 212-218.

Faculty of Technical Sciences, "Integrisane akademske studije tehnika i informatika," 2013, http://www.ftn.kg.ac.rs/akreditacijaFTN/index.php?sp=IS_TI

G. Demartini, I. Enchev, J. Gapany, and P. Cudré-Mauroux, "The Bowlogna Ontology: Fostering Open Curricula and Agile Knowledge Bases for Europe's Higher Education Landscape," Semantic Web - Interoperability, Usability, Applicability, vol. 4, no. 1, 2013, pp. 53-63.

I.F. Cruz, W. Sunna, N. Makar, and S. Bathala, "A Visual Tool for Ontology Alignment to Enable Geospatial Interoperability," Journal of Visual Languages and Computing, vol. 18, no. 3, 2007, pp. 230-254.

J. Euzenat, and P. Shvaiko, Ontology Matching, Springer-Verlag, Berlin-Heidelberg, 2007, p. 333.

J. Li, J. Tang, Y. Li, and Q. Luo, "RiMOM: A Dynamic Multistrategy Ontology Alignment Framework," IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 8, 2009, pp.1218-1232.

J.T. Fernández-Breis, D. Castellanos-Nieves, J. Hernández-Franco, C. Soler-Segovia, M. C. Robles-Redondo, R. González-Martínez, and M. P. Prendes-Espinosa, „A semantic platform for the management of the educative curriculum," Expert Systems with Applications, vol. 39, no. 5, 2012, pp. 6011-6019.

M. Jaro, "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, 1989, pp. 414–420.

M. Mandić, M. Segedinac, G. Savić, and Z. Konjović, "IEEE RCD standard based ontological modeling of Computer Science curriculum," Proceedings of the 3th International Conference on Information Society and Technology, 2013, pp. 189-285.

M. Mandić, Z. Konjović, and M. Ivanović, "Ontological Model of the Standardized Secondary School Curriculum in Informatics," Proceedings of the 5th International Conference on Information Society and Technology, 2015, pp. 363-367.

M. Mandić, Z. Konjović, and M. Ivanović, "Platform for computer-aided harmonization of informatics curricula," Acta Polytechnica Hungarica, vol. 13, no. 3, 2016, pp. 159-179.

R.Heer, "A Model of Learning Objectives–based on A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives," Center for Excellence in Learning and Teaching, Iowa State University, 2012. http://www.celt.iastate.edu/teaching-resources/effective-practice/revised-blooms-taxonomy/

W. Winkler, "The state of record linkage and current research problems, tech. report 99/04," Statistics of Income Division, 1999, Internal Revenue Service Publication

Y. R. Jean-Mary, E. P. Shironoshita, and M. R. Kabuka, "Ontology matching with semantic verification," Web Semantics: Science, Services and Agents on the World Wide Web, vol. 7, no. 3, 2009, pp. 235–251.

Y.L. Chi, "Ontology-based curriculum content sequencing system with semantic rules," Expert Systems with Applications, vol. 36, no. 4, 2009, pp. 7838–7847.