КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (31), 2025
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2025.31.1043
UDK 004.4, 004.6, 004.9

Tkachenko Kostiantyn
PhD of economical sciences, associate professor, a
ssociate professor at the department of software engineering
State University "Kyiv Aviation Institute", Kyiv, Ukraine
associate professor at the department of information technologies
National Transport University, Kyiv, Ukraine.
ORCID: 0000-0003-0549-3396
tkachenko.kostyantyn@gmail.com

Tkachenko Olha
PhD of physical and mathematical sciences, associate professor,
associate professor at the department of information technologies
National Transport University, Kyiv, Ukraine.
associate professor at the department of computer science
Borys Grinchenko Metropolitan Kyiv University, Kyiv, Ukraine
ORCID: 0000-0003-1800-618X
oitkachen@gmail.com

Tkachenko Oleksandr
PhD of physical and mathematical sciences, associate professor,
associate professor at the department of computer systems software
National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.
associate professor at the department of information technologies
National Transport University, Kyiv, Ukraine.
ORCID: 0000-0001-6911-2770
aatokg@gmail.com

INTELLIGENT LEARNING SYSTEMS: TRANSFORMING TABULAR DATA OF EDUCATIONAL CONTENT IN THE ONTOLOGICAL MODEL

Abstract. The paper proposes to automate the process of forming ontologies based on the analysis and transformation of tabular data of complex structures, which is educational content in modern intelligent (or information system with elements of intellectualization) learning systems. An approach is presented that ensures the restoration of the semantics of tabular data, as well as conceptualization and the formalization of the tabular content of educational content in the form of an ontology. The main stages of the approach are given. The described tools can be used to solve practical problems of providing educational content in information learning systems. The educational content presented in the form of tables of the system database was used as the initial data. The proposed approach is advisable to use for prototyping subject ontologies. This article examines the creation of intelligent learning systems, taking into account all the opportunities offered by transforming tabularly defined educational content into corresponding multi-level ontological model. Hroposed ontological approach enables the implementation of learning processes by supporting the shared use of common educational content stored in tabular form using ontological model. This article analyzes the main challenges of representing complex tabular data in ontological model used to describe educational content. It is proposed to use hybrid ontology construction methods, table analysis, and logical and heuristic approaches to defining entities and key columns for analyzing and processing such data. Ontology can be constructed for a specific topic of a studied academic discipline using similar data from various sources. This approach will allow for the reflection of diverse conceptual formulations, take into account diverse perspectives, and eliminate the subjectivity of a specific source. Using semantic dictionaries allows for the unification of virtually any dataset into a single ontology for further solving various problems related to the educational process.

Keywords: tabular data; data transformation; ontology; ontological model; information learning system; information learning system with elements of intellectualization; educational content

INTRODUCTION

Modern intelligent (or information system with elements of intellectualization) learning systems are actively used in the management and control of all learning processes (in particular, the provision of educational content, testing and monitoring of students' knowledge) [1], [2], [3].

To build this kind of system, ontologies [4], [5], [6] can be used, designed to accumulate and transfer knowledge about the subject area under consideration, with their vertices (nodes) representing objects of interest, and edges representing relationships between these objects [7], [8]. Ontologies can be represented by the following formula [9], [10]:

$$O = < X, R, F >,$$

where X is a set of concepts (elements, terms) of the subject area, which is represented by the ontology O;
R is the set of relations between the concepts of the subject area under consideration;
F is a set of interpretation (axiomatization) functions defined on concepts and/or relations of the ontology O.

Ontologies can be divided into:

- meta-ontologies;
- subject-oriented (subject) ontologies.

Subject ontologies support effective knowledge retrieval. Their use in the construction of intelligent (or information system with elements of intellectualization) learning systems allows for effective knowledge structuring and identification of hidden connections and dependencies between different concepts, which can be useful for making decisions on specific learning (teaching, training) processes (including individualization of learning for the specific student or use of gamification elements).

Developing ontologies in intelligent (or information system with elements of intellectualization) learning systems is a labor-intensive task, the solution of which requires using and processing large amounts of data obtained from various information sources (e.g., databases, educational content, web resources).

The primary trend here is the use of various information sources. One such source is spreadsheets [11], [32], [33]. Tables are heterogeneous in their structure and are not accompanied by explicit semantics necessary for automatic interpretation of their content.

This complicates the active, practical use of tabular data in automatic/automated modes. An approach to automated filling of ontologies with entities based on table analysis (tabular data) was proposed in [12], [31]

A unique feature of this approach is the ability to automatically restore the semantics of tabular data based on using a method that combines machine learning, vector representations and intuitive heuristics.

The construction and implementation of intelligent systems based on the formalization, and reuse of data and knowledge is a promising direction for the practical application of artificial intelligence methods in software systems. Such systems are based on a formalized representation of knowledge about a subject area, for example, in the form of an ontology.

To build the theoretical foundations of databases, formal methods are often used based on [13], [31]:

- algebraic approach (including the use of multi-sorted algebras to describe databases

and knowledge bases (based on the use of abstract data types (used for example, to represent the concepts of a specific subject area), the study of equivalence and symmetry of relations for algebraic modeling of operations);

– set theory;
– predicate logic.

Tabular data is one of the most accessible and widespread ways of representing and storing information, characterized by a wide variety and heterogeneity of layouts, styles and content, while remaining a source of structured subject knowledge.

Data representation through ontology supports and provides convenient and transparent data visualization, fast navigation, important connections necessary for data analysis and identification of new knowledge [14].

Research of ontological modeling has focused mainly on declarative ontologies: subject area ontologies, general ontologies, and task ontologies have not received enough attention yet.

Task ontology is the result of development of methods of task analysis in defining and formalizing factors influencing the process of problem solving by an expert. Ontologies facilitate formalization of concepts and relations of a task.

Task ontologies are characterized, in particular, by the fact that:

– They are built separately for classes of similar tasks.
– The following are used:
  - the concept of the goal of solving a problem and its formalization;
  - the concept of an action and ensures execution/modeling of an action;
– The execution of a model built based on a task ontology is implemented.

When a task ontology is built, its formalized representation occurs, based on close interaction with the subject area expert who creates and validates the ontology.

In the modern globalized information space, the data encountered by their users have different presentation formats.

The most typical (standart) format used when working with a large amount of structured data is database tables. This format is well suited for machine processing, studying, and using large-scale data, but with “manual” processing of data presented in tables, this is a very labor-intensive process [15].

Ontologies allow to combine heterogeneous information from different sources and to provide its representation using formal standardized means of knowledge modeling.

Ontology can be used for effective representation of knowledge and translation of knowledge into a form suitable for interpretation both by corresponding software products (information and intelligent systems, web services, etc.) and by people [15].

Research into creating ontologies and improving the efficiency of these processes has been conducted for many years. As a result of this work, the following should be highlighted:

– ontology development projects [7] [35], [36] (e.g., KACTUS [34], On-To-Knowledge (fig.1) [37], NeOn [38], [39]);

– software tools for forming ontologies – the so-called ontology editors [40], [41] (e.g., Protégé [26], Fluent Editor [42], OntoStudio [43], ONTOedit [44], WebOnto [45]).

ISSN 2663 - 4023

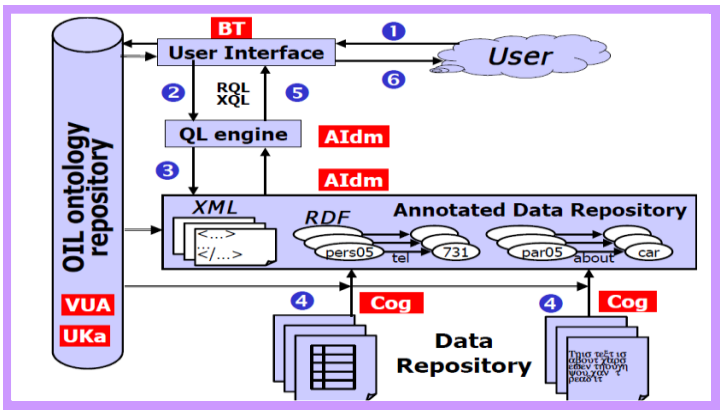

Figure 1. The On-To-Knowledge toolset for ontology-based knowledge management [37]

However, despite the results obtained, even today, developing ontologies remains a complex, creative, and labor-intensive task.

Automation of forming ontologies based on various information sources (e.g., professionals, databases, texts, conceptual models) allows using vast volumes of accumulated information presented in various formats.

The work focuses on a detailed examination of such an information source as tabular data stored in the corresponding database tables. Tabular data with different (often quite complex) structures are of greater interest.

To work with tabular data, it is proposed to use ontological modeling (creation of ontologies based on the analysis and transformation of data extracted from DB spreadsheets) presented in CSV format.

The main problem of the ontological form of information presentation is the complexity of its implementation; therefore, approaches to programmatic display of tabular data of a complex structure in an ontological model are proposed.

The proposed approach to creating ontologies uses tabular data presented in a relational database, which allows for the unification of data necessary to form an ontology.

Tabular data contains structured information in the form of rows and columns, where a column displays the name of an object (in ontology terminology – the name of an individual) or a specific, particular attribute (in ontologies – the data property), and each row – description of the specific object by its properties.

In this representation, the column headers (in the top row of headers) contain the names of attributes, and the rows represent sets of attribute values with different data types.

An ontological model is a formal model of knowledge describing entities of the corresponding problem area, their properties and relationships between them.

ONTOLOGICAL MODEL OF TABULAR DATA REPRESENTATION

Ontological modeling of processing processes and representation of tabular data of different (often complex) structures requires appropriate justification in a formal model. To build a formal ontological model, a toolkit similar to system algebras was used [6], [13], [18], [29], [30].

Subject areas whose tabular data are used and described by the corresponding ontological model can be represented as follows:

$$SA = \{SA_1, SA_2, \ldots, SA_n\},$$

where SA_i is the set of subject area objects that concretize some specific concept T_i (classified as instances of this concept T_i).

Sets SA_1, SA_2, ..., SA_n are carrier sets for n multi-sorted algebras. For ontological modeling, we will use:

Domains of subject area's concepts (*DC*). For each set SA_i, we define an abstract data type DC_i of its elements, which can be represented as:

$$DC_i = <Name_{DC\_SAi}, \sum_{A\text{-}SA}, DC_{SAi}>,$$

where $Name_{DC\_SAi}$ is the name of the abstract data type DC_i, $\sum_{A\text{-}SA}$ is the signature of the multi-sort algebra for the subject area *SA*, DC_{SAi} is the set of basic (essential, fundamental) relationships of the DC_i type.

The name of the abstract data type DC_i is not specified (not defined) when an arbitrary type is used to describe the data. However, if a specific data type must be specified for processing and presenting tabular data, then the $Name_{DC\_SAi}$ parameter is mandatory.

The set SA_i can be represented as a union of sets that has the form:

$$SA_i = \bigcup_{j=1}^{k} SA_{ij}$$

Where SA_{ij} is the set of software objects that specify some specific concept T_i, have one standart specific data type, and *j* is the number of the data type in the set of software data types *SA* ($j = 1, \ldots, k$).

Data types DC_i form the set *DC* of concept domains corresponding to the concepts (entities) of the software object.

$$DC = \bigcup_{i=1}^{n} DC_i$$

Concepts are divided into classes and class values. In this case, classes can form a hierarchy, i.e. the value of a class can be another class (subclass), for example, the class "EDUCATIONAL_CONTENT" can include as the value the subclasses "TEXT_MATERIAL", "XML_DOCUMENTS", "PDF_DOCUMENTS", "GAMIFICATION_ELEMENTS", "TEST". Relationships between concepts are divided into classification – between classes and subclasses; structural, describing the interaction of classes.

Attribute domains (*DAt*) of subject area's objects are defined based on the set of attribute values available for subject area *SA*.

The elements of this set have the form:

$$<ID_{At}, sign_{At}>,$$

where ID_{At} is the attribute identifier, and $sign_{At}$ is the attribute value (with the possible taking into account the semantics of the attribute).

The operations defined on the *DAt* attribute domain are union, substitution, deletion, and interpretation.

$$DAt = \bigcup_{l=1}^{m} DAt_l$$

where DAt_l is the attribute domain of the *l*-th entity ($l = 1, \ldots, m$).

Logical domain (DL_{SA}) includes expressions (for example, simple logical expressions, predicates), the result of calculating the values of which will be one of the elements of the set {*true, false*}.

The operands of expressions can be objects of other domains.

Elements of the logical domain will be interpreted as constraints that can be imposed, for example, on attribute values. Operations on DL_{SA} objects can be conjunction (*and*), disjunction (*or*), negation (*not*) and interpretation (*interp*).

$$DL_{SA} = \bigcup_{l=1}^{m} DL_{SA_l}$$

where DL_{SA_l} is the logical domain for attributes of the *l*-th entity ($l = 1, \ldots, m$).

Domain of concepts with attributes (DC_{At}) is defined on a set whose elements have the form:

$$< DC_i, DAt_l, DL_{SA_l} >$$

Each constraint DL_{SA_l} is the logical expression with operands belonging to the domain DAt_l. Operations on elements of domain DC_{At}: merging and splitting entities {*merge, split*}. The merging entity operation is the formation of the new entity as the standart (typical) set of properties and constraints of the operand entities.

The splitting entity operation is the reverse of merging. An instance of the DC_{At} domain is the tuple of the corresponding facts.

Relation domain (*DR*). Each object of this domain is represented by the structure that contains the Cartesian product of algebraic data types from the domain DC_{At}, the type from the *DAt* attribute domain, and the set DL_{SA} of constraints.

The operations of merging relations, separating, and substituting {*merge, split, substitute*} are defined over relations. These operations are treated similarly to similar operations from the domain *DC*. In the substitution operation, instead of one entity of the domain DL_{SA}, we substitute the relation, which is treated as an entity with attributes.

The ontology that is formed based on tabular data of the educational content of the information training system can be represented as a tuple of the form:

$$Ont = <DC_{At}, DR, DL_{SA}>,$$

where DC_{At} is the finite set of concepts (classes, concepts) of the ontology with their properties (attributes); *DR* is the finite set of relations (connections, correspondences) between concepts; DL_{SA} is the finite set of interpretation functions (constraints, axioms).

The main data elements in the ontology *Ont* are:

– Classes – categories or types of objects, for example, the classes "Educational course" or "Specialty".

– Instances (individuals) – specific representatives of classes, for example, for the class "Educational course" – "Software Quality and Testing", and for the class "Specialty" – "Software Engineering".

– Annotations – localized tags, comments, and other metadata of entities;

– Data properties – entities that characterize instances, acquiring values of a specific type, for example, the value of the property "Semester of study" for the individual "Software Quality and Testing" – "8" (can be represented as integer or text value).

Using data properties, you can represent various essential information about an object, for example, the value of the property "Type" for the individual "Software Quality and Testing" – "Required".

Object properties (also the data element in the ontology *Ont*) are relationships between entities. For example, the "C++" or “C#” language is an extension of the "C" language [16].

APPROACH THE CREATION OF ONTOLOGIES BASED ON TABULAR DATA

Of particular interest are approaches and software tools for extracting related data and ontologies from data tables (tabular data with arbitrary structure). There are software tools that allow, in particular:

– extract RDF-triplets from such tables (for example, csv2rdf4lod, RDF123, Spread2RDF);

– form OWL ontologies (for example, Any2OWL, Excel2OWL, Owlifier);

– describe transformations of tabular data into sets of related data or ontology using special (unique) object-oriented languages (for example, XLWrap, Mapping Master, RML, PEARL);

– use languages and tools based on the extension of the SPARQL query language (for example, XSPARQL, SPARQL-Generate, Tarql).

– Despite the achieved results of transformation of tabular data, the above tools have some disadvantages:

– The use of tabular data models with mixed physical and logical schemes causes:

- Limiting the use of many tools for processing arbitrary electronic data tables.
- Complicating the understanding and application of some solutions based on object-oriented languages (for example, Mapping Master, SPARQL-Generate, etc.), since the user needs to know the syntax of these languages and the rules for constructing correspondences between elements of two formats.
- Does not provide a visual support of conversion, etc.

– Weak support for the formation of ontologies based on the analysis and transformation of a set of tabular data belonging to the same subject area, i.e., as a rule, the tools support a single transformation of the table into an ontology (or some of its fragments) and do not assume the unification of the obtained ontologies for processing the entire set of electronic tables.

– The existing tools are primarily oriented toward programmers and do not assume use by other categories of users (for example, experts in the subject area).

– Lack of quality control of received ontologies and sets of related data.

The analysis and processing of tabular data of the complex structure, obtained from different sources and planned to be used in the future for the enrichment of ontologies, require the solution of such tasks [16].

1. The need to determine which column contains entity identifiers (proper names, names, etc.). Usually, this is the first or second column on the left, but it is possible that the table may contain its identification system (simple numbering or unique indexing).

2. Provision of a reading label for each of the entities (although the unique identifier of the entities of the ontology may look like a unique index (for example, "E00121" may correspond to the specialty "Software Engineering"), it is important that the ontology must facilitate the understanding of the data (their semantics)).

3. Using proper names for identification (the problem is that these names can be repeated).

4. The actual types of entities are an ambiguous concept, and they must be reflected in the ontology using classes. The problem is that these types are unknown in advance, and each entity can belong to more than one type.

5. Even initially, complete and correct data, due to the peculiarities of transformation/formatting, may contain so-called "noise" (empty cells, invalid links, extra spaces, line breaks, etc.).

6. The same essence can be represented differently even in one data set. For example, the entity denoting the paradigm of object-oriented programming can be represented as:

"Object-oriented programming", "Object-oriented programming", "Object-oriented", "OOP", etc. d.

7. The proper name can change. That is, even in unified data, the same entity can be described differently, in different places.

The following ways are offered to solve the above problems:

1. *Composite key*. Accompanying information can provide additional context to unambiguously define the essence and prevent duplication or loss of factual links.
2. *Determination of parameters manually*.
3. *The use of dictionaries* allows you to configure the program (in particular, for the translation, processing, and use of tabular data), for example, to define the list of "garbage" or, on the contrary, mandatory data, abbreviated, alternative names, etc.
4. *Identification of logical data*. With a small number of possible values and the keywords and symbols (Yes/No, True/False or +/-), the column can be identified as logical and its value can be used as belonging/non-belonging to the class presented in its header.
5. *Identification of the central column* (with entity names). Most tables contain a column whose values are the names of the described entities. To determine such a column, you can use the method when such a column is defined as the leftmost column with the maximum number of unique (non-numeric) values.
6. *Step-by-step analysis of tables* (tabular data of a complex structure), supplementing each subsequent stage (for example, metadata analysis, vertical analysis, horizontal analysis) with the findings of the previous one.
7. *Using* the previously *created ontology as the "standard"*. An ontology created manually or based on data from a verified source can be a "standard" against which the generated ontology will be compared. Having reached a complete match by improving the algorithm, you can continue testing by gradually degrading the source table to check the algorithm's robustness to errors.
8. *Post-analysis*. Analysis of textual information (labels, comments, description, value of data properties) to create object relationships between entities and additional ontology enrichment.
9. *Combining data sets* when a dictionary of <key-value> pairs is used as a link between two or more data sets. This approach can be a tool for creating ontologies from two or more tabular data sets.
10. *Adding data to* the existing *ontology*. Using an existing ontology file, you can create a knowledge graph for tabular data. For example, the reference ontology can be used for the validation of lower-quality tabular data, which will supplement the ontology with new knowledge [15], [16].

In this way, it is possible to combine different data sets, creating unique ontologies (networks of ontologies) for solving various practical, for example, interdisciplinary problems.

INTEGRATION OF TABULAR DATA SEMANTICS IN ONTOLOGICAL MODELING

Integration of tabular data semantics involves the formation of a single content space for the perception, interpretation, and application of data regardless of their presentation format and structure. The purpose of such processes is the same interpretation of all data elements obtained from various sources and components of a specific unified information structure.

One of the problems in this process is the formation of criteria for the semantic integration of data sets, which can be used to assess the possibility or impossibility of combining their content and resolve semantic conflicts between data sets subject to integration in the process of ontological modeling educational content of information learning system (based on tabular data) [6], [8]. Among such semantic conflicts, the following can be distinguished in particular [17], [18]:

– An ambiguity conflict occurs when two concepts look the same but are essentially different, for example, the concept of "obligations".

– The metrics conflict occurs when the same quantities are measured by units of different systems, for example, using different measures of weight (kg or pounds) to determine the weight of the same product.

– Conflicts of names occur when the naming systems of concepts and objects are fundamentally different.

– This often manifests itself in the appearance of synonyms and homonyms.

Semantic integration of tabular data based on ontologies. One of the approaches to the integration of tabular data semantics is integration using ontological models, which involves the use of a thesaurus and metadata and takes into account a wide range of different aspects of tabular data semantics.

The ontology should be considered as a holistic formalized specification of a specific software product, which should ensure the exact interpretation of knowledge about this software product. When integrating data, the object of the description presented by means of an ontology is a specific information resource [19], [20], [21].

Therefore, it is advisable to talk about data ontology (including tabular data of different nature, obtained from different sources, and having both simple and composite structure).

Semantic integration of tabular data based on ontologies involves defining:

– the procedure for constructing and applying ontology (the role of ontology as a means of describing the semantics of tabular data affects the methods of its formation and presentation);

– methods of displaying the semantics of tabular data in an ontology;

– procedures for applying equality (identity) of concepts (in integration processes, depending on the specifics of the data to be integrated, equality may have various interpretations; in particular, such options as exact equality, partial coincidence, equivalence, similarity, etc. are used).

Prototyping ontologies based on analysis and transformation of tabular data. Let us describe the operator for transforming arbitrary tabular data of a complex structure into an ontology as follows:

$$Transformation: Tabular\ Data \rightarrow Ontology$$

For more convenient formal description of the step-by-step process of forming an ontology from tabular data (of educational content), let us present this transformation operator as follows:

$$Tr: TD \rightarrow Ont$$

that tabular data (for a more precise display in the corresponding ontology) should be presented as tables in CSV format.

The ontology being formed is an OWL ontology. The Tr operator has its reflection at each stage of the transforming of arbitrary tabular data of a complex structure into an ontology.

$$Tr = \bigcup_{i=1}^{3} Tr_i$$

ISSN 2663 - 4023

where:

– $Tr_1: TD \to TD_{rel}$, TD_{rel} are the original tabular data presented (transformed) in relational form, Tr_1 is the set of rules for converting the original tabular data of the complex structure, presented in CSV format, into the relational format;

– $Tr_2: TD_{rel} \to OM_{SA}$, where OM_{SA} is the ontological model of the subject area,

– $OM_{SA} = OM_{Con} \cup OM_{Ax}$, where OM_{Con} is ontological model of the subject area at the conceptual level (level of the concepts), OM_{Ax} is the ontological model of the subject area at the axiomatic level;

Tr_2 is the set of rules for transforming complex tabular data (of educational content) presented in relational form, into the corresponding subject area the ontological model (which is a description of the subject area at the level of concepts and axioms);

– $Tr_3: OM_{SA} \to Ont$, where Tr_3 is the set of rules for transforming the ontological model into the OWL-ontology code.

Description of the stages of the proposed approach. The formation of ontologies based on the analysis and transformation of tabular data of complex structure involves the following sequence of actions (stages):

1. Analysis and transformation of arbitrary spreadsheets into a uniform canonical form
2. Recognition of named entities and determination of cell types
3. Obtaining ontology fragments (extraction of the ontological schema and specific facts)
4. Aggregation of ontology fragments
5. Generation of ontology code in OWL format

The transformation performed at stage 1 includes the following phases: recognition, role (functional), and structural analysis. The tabular data used at this stage are presented in the following form:

$$TD = <D_{rec}, R_{head}, Col_{head}>,$$

where:

– D_{rec} is the complete data fragment describing specific data values (these are fields (columns) of records in the tables of the corresponding databases)), belonging to the same data type supported by the corresponding databases management system.

– R_{head} is the set of database table row headers, i.e.

– Col_{head} is the set of column headers of the database tables, i.e.

$$D_{rec} = \bigcup_{i=1}^{n} \bigcup_{j=1}^{m_i} D^i_{rec_j}$$

Where $D^i_{rec_j}$ is the *j*-th consistent data fragment of the *i*-th database table; *i* is the database table number (*i = 1, 2, …, n*), *j* is the data fragment number in the *i*-th database table (*j =1, 2, …, m_i*, where m_i is the number of consistent data fragments in the *i*-th database table).

$$R_{head} = \bigcup_{i=1}^{n} \bigcup_{l=1}^{p_i} R^i_{head_l}$$

Where $R^i_{head_l}$ is the header of the *l*-th row of the *i*-th database table; *i* is the database table number (*i = 1, 2, …, n*), *j* is the data row number in the *i*-th database table (*l =1, 2, …, p_i*, where p_i is the number of data rows in the *i*-th database table),

$$R^i_{head} = \{R^i_{head_1}, R^i_{head_2}, \dots, R^i_{head_{p_i}}\}$$

ISSN 2663 - 4023

$$Col_{head} = \bigcup_{i=1}^{n} \bigcup_{k=1}^{d_i} Col^{i}_{head_k}$$

where $Col^{i}_{head_k}$ is the header of the *k*-th column of the *i*-th database data table; *i* – database table number (*i = 1, 2, …, n*), *k* – data column number in the i-th database table (*l =1, 2, …, d_i*, where d_i is the number of data columns in the *i*-th database table),

$$Col^{i}_{head} = \{Col^{i}_{head_1}, Col^{i}_{head_2}, \dots, Col^{i}_{head_{d_i}}\}$$

The values in the header block cells are separated by a unique separator character, which represents hierarchical relationships between the headers.

At this stage, pre-preparation of the tabular data presented in canonical form is performed for further processing, including, in particular:

– correction of so-called "broken" Unicode characters;
– removal of various "garbage" character values, except for letters and numbers;
– decoding acronyms; removing multiple spaces;
– identifying and removing units of measurement, etc.

At the stage 2, the procedure of extraction and recognition of named entities contained in the cells of the canonical table of data base (resource of modern intelligent (or information system with elements of intellectualization) learning system) is carried out.

For this, the libraries for natural language processing is used (e.g., Stanford CoreNLP, Stanford Named-Entity Recognizer (Stanford NER) [22], [23], [24]. Stanford NER:

– marks words in the text that are names of objects;
– defines a set of classes of named entities.
– The determined typed cells are divided into:
– cells with named entities (named-entity cells);
– cells with literal values (literal cells).

Such cell typing allows us to classify tabular data into different ontological levels (the level of class properties and specific instances) at the next stage of transformation.

At the stage2, specific instances (axiomatic level of ontology) are extracted from the tabular data presented in canonical form based on D_{rec}. In this case, only those cell values that contain named entities are taken into account.

The task of the stage 3 is to obtain ontological fragments in the form of a set of classes, their relations and property-values, and specific instances (facts) describing a particular subject area, based on the analysis and transformation of tabular data presented in canonical form.

To obtain ontology fragments, it is first necessary to extract an ontological schema (the terminological level of the ontology) from the prepared tabular data based on the row and column headers.

In this case, heuristic rules for transforming tabular data are used. Such rules have also been developed for the situation when the class structure is formed based on the labels in R_{head}.

The obtained header relationships are interpreted as relations between classes. The values for the property-values are established based on the records from D_{rec}, which are defined as literal at the stage 3.

The primary result of this stage is fragments of the ontological model. These fragments must be aggregated, including operations to clarify the names of classes, their properties and relations, their merging and splitting.

The task of the stage 4 is to combine the obtained ontology fragments into a single aggregated ontology.

This model is intended for the unified presentation and storage of knowledge extracted from different information sources. Automatic aggregation of ontology fragments uses the following rules:

– Classes with the same names are combined, forming the typical set of value properties, object properties, and instances.

– Classes with the same names and property structure are deleted.

– Classes with similar names are combined.

– The obtained fragments of the ontological model *Ont* can describe the same objects or processes. It is proposed to use the term comparison method to determine the similarity between two class names.

– Creation of new relationships between classes if there are classes and value properties of the same name. In this case, a new class with the name of the value property is created, and the value property of the same name is deleted.

– Duplicate object characteristics between classes are deleted.

– Duplicate value characteristics are deleted.

Duplicate instances are deleted.

The task of the stage 5 is to generate the ontology code in the OWL format [25].

The OWL code of the ontology is based on the obtained fragments of the ontology models. The generated OWL code of the ontology can be modified and supplemented using the ontology editor Protégé [26].

Semantic interpretation of data tables involves [27], [28]:

– annotating cells – matching cell values with entities (instances of classes) from the ontology;

– annotating columns – matching individual table columns with semantic types (classes) from the ontology;

– annotating relationships between columns – matching relationships between columns with properties from the ontology;

– annotating the table – matching the entire data table with the specific class from the ontology (defining the table topic).

Development of Subject Ontologies Based on Semantic Annotation of Tabular Data, This work proposes an approach to automatically extracting specific entities (facts) from tables and filling the corresponding ontology with them.

Unique feature of this approach is the ability to support automated restoration of table semantics based on the subject area model (ontology at the terminology level). Due to this, it is possible to specify explicit semantic annotation for individual table elements (columns and relationships between them) and extract specific entities from cells. In this case, it is possible to solve two problems of semantic interpretation of tables: annotating columns and annotating relationships between columns.

The approach has several limitations, in particular, it is focused on processing only relational tables presented in CSV-format. The proposed approach implements semantic annotation of columns and relationships between them, which consists of matching columns with specific types of characteristics, finding the most suitable type of concept based on them, and identifying of relationships between specific types of concepts.

Approach stages:

– *Table preprocessing*. At this stage, named entity recognition (NER) is performed for each cell in the source table. Specific NER labels of named entities are assigned to each cell in the source table, characterizing the data it contains. Depending on the assigned NER label, mentioned facts and value facts corresponding to the characteristic value type defined in the

subject area model are automatically extracted from the cells. At this stage, characteristic facts and concept facts can also be extracted from the cells.

– *Search for candidate types*. For each column, a set of candidate characteristic types is formed, obtained from the subject area model based on specific mention facts and value facts. Columns for which facts were not extracted in the previous step are excluded from subsequent table processing.

– *Semantic annotation of columns*. At this stage, the most suitable feature type from the set of candidates is selected for assignment to the column. This is done using a method consisting of a combination of the following heuristics:

- Majority voting is a basic solution, which consists of assigning the most suitable type from the set of candidates to the column based on direct inference from the feature facts that were extracted for the column cells. Then, the frequency of occurrence of each candidate type is calculated.
- Header similarity is a lexical comparison of the column header with the names of feature types from the set of candidates. If the column contains concept facts, the header name is compared not with the names of feature types from the set of candidates, but with the names of concept types that are associated with these feature types.
- Feature grouping is a heuristic based on the fact that a table may have one or more columns in which some concept facts and feature facts have already been extracted. For each such column, the number of possible characteristics that are located in other columns and relate to this concept is calculated. Then the column with the maximum number of characteristics is determined.

Based on these heuristics, a final assessment is determined that a specific type of characteristic from the set of candidates is the most suitable for annotating a table column.

– *Fact extraction*. Based on the established column annotations, new facts-concepts, facts-values, facts-mentions, facts-characteristics of concepts are extracted from the table. In this case, the extracted facts-mentions include the value of the entire cell. Facts are extracted row by row from left to right. Facts-characteristics are created only for the leftmost categorical column in the table.

If the same type of characteristic is defined as an annotation for several categorical columns in a table (for example, if a table has two columns with persons, and all other columns are defined as some characteristics of a person, then only for the facts-concepts from the first column the corresponding characteristics are created).

In this case, identifying characteristics (names) are always extracted. Based on the extracted facts-concepts, all possible facts-connections defining the relationships between two concepts are also extracted from the table row by row.

Ontologies developed based on tabular data (of educational content) can be described by various means. It should be taken into account that any ontology defines terms (concepts) and specifies logical connections between them. The semantics of the description of terms and connections is based, in particular, on:

– the formalization, i.e. description of objects of the subject area using uniform, strictly defined samples (concepts, terms, models, etc.);

– use of the limited number of basic concepts (entities), based on which all other concepts are constructed;

– internal completeness;

– logical consistency.

Ontologies are characterized by internal unity, logical interconnection and consistency of the concepts used. The process of data integration implies integration of their structure, syntax and semantics. Semantic integration of tabular data is based on the construction and processing of ontologies that ensure their joint use in the forming a single semantic space of integrated tabular data.

An important aspect of semantic integration is determining the possibility of joint use of input tabular data sets.

Semantic integration criteria provide the ability to determine and coordinate the constituent ontologies of input tabular data sets.

Semantic data integration criteria can be formulated as a sequence of requirements for pairwise coordination of data ontology elements. Fulfillment of the entire set of requirements allows us to conclude that it is possible to integrate two data sets at the level of their content with the receipt of a semantically correct result.

CONCLUSIONS

This article analyzes the main problems of displaying tabular data of complex structure in an ontological model used to describe educational content in intelligent learning systems.

It was proposed to analyze and process such data, including of hybrid methods for constructing ontologies, step-by-step analysis of tables, logical and heuristic approaches to identifying entities and key columns.

The using of the proposed approach allows you to build an ontology on a specific topic of studying an academic discipline, using the same type of data from different sources. This approach will reflect different formulations of concepts, take into account different points of view and eliminate the subjective point of view of the particular source.

Using semantic dictionaries, you can combine almost any data set into a single ontology for further solving various problems related to learning processes.

Ontologies allow you to systematize knowledge, making complex concepts more accessible for understanding. Therefore, one of the considered methods of practical use of the methodology is the organization and management of educational processes at the university.

The proposed approach provides simplicity and flexibility in working with tabular data, taking into account considering their structure and context.

Based on the results of theoretical research, a software product has been developed that can process heterogeneous tabular data, identify, classify ontological entities, determine and establish their attributes, create ontological links and facilitate the identification of new knowledge.

The practical results of the research confirm the relevance of developing mapping of tabular data of complex structure into an ontological model, reducing the time and resource costs of building ontologies and facilitating more efficient analysis of large volumes of information (e.g., educational content of intelligent (or information system with elements of intellectualization) learning systems).

As a result of the research, key problems were identified and solutions were proposed. The practical significance lies in creating of tools for automating data processing and analysis and identifying new knowledge based on the use of ontologies, improving educational and management processes.

REFERENCES

1. Liu, G. Z. (2017). A key step to understanding paradigm shifts in E-learning: Towards context-aware ubiquitous learning. *British Journal of Educational Technology, 41*(2), E1–E9.
2. Scherer, M. U. (2016). Regulating artificial intelligence systems: Risks, challenges, competencies, and strategies. *Harvard Journal of Law & Technology, 29*(2).
3. Stancin, K., Poscic, P., & Jaksic, D. (2020). Ontologies in education – State of the art. *Education and Information Technologies, 25*, 5301–5320. https://doi.org/10.1007/s10639-020-10226-z
4. Munira, K., & Anjumb, M. S. (2017). The use of ontologies for effective knowledge modelling and information retrieval. *Applied Computing and Informatics*. https://doi.org/10.1016/j.aci.2017.07.003
5. Bouarab-Dahmani, F., Comparot, C., Si-Mohammed, M., & Charrel, P. J. (2015). Ontology-based teaching domain knowledge management for e-learning by doing systems. *The Electronic Journal of Knowledge Management, 13*(2), 155–170.
6. Tkachenko, K. O. (2022). Using ontological modeling by intellectualization of learning processes. *Digital Platform: Information Technologies in Sociocultural Sphere, 5*(2), 261–270. https://doi.org/10.31866/2617-796X.5.2.2022.270130
7. Alfaifi, Y. H. (2022). Ontology development methodology: A systematic review and case study. In *Proceedings of the 2nd International Conference on Computing and Information Technology (ICCIT)*. https://doi.org/10.1109/ICCIT52419.2022.9711664
8. Tkachenko, O., Tkachenko, K., & Tkachenko, O. (2020). Designing complex intelligent systems on the basis of ontological models. In *Proceedings of the Third International Workshop on Computer Modeling and Intelligent Systems (CMIS-2020)* (pp. 266–277).
9. Gavrilova, T. (2014). Building collaborative ontologies: A human factors approach. In P. Diviacco, P. Fox, C. Pshenichny, & A. Leadbetter (Eds.), *Collaborative Knowledge in Scientific Research Networks* (pp. 305–324). IGI Publishing.
10. Sowa, J. (n.d.). *Building, sharing and merging ontologies*. http://www.jfsowa.com/ontology/ontoshar.htm
11. Bonfitto, S., Casiraghi, E., & Mesiti, M. (2021). Table understanding approaches for extracting knowledge from heterogeneous tables. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 11*(4), e1407. https://doi.org/10.1002/widm.1407
12. Liu, J., Chabot, Y., & Troncy, R. (2023). From tabular data to knowledge graphs: A survey of semantic table interpretation tasks and methods. *Journal of Web Semantics, 76*, 100761. https://doi.org/10.1016/j.websem.2022.100761
13. Burov, E. V. (2012). *Conceptual modeling of intelligent software systems*. Lviv Polytechnic Publishing House. (in Ukrainian)
14. Jiménez-Ruiz, E., Hassanzadeh, O., Efthymiou, V., Chen, J., & Srinivas, K. (2020). Resources to benchmark tabular data to knowledge graph matching systems. In *The Semantic Web. ESWC 2020* (Vol. 12123). Springer. https://doi.org/10.1007/978-3-030-49461-2_30
15. Gomez-Vazquez, M., Cabot, J., & Clarisó, R. (2024). Automatic generation of conversational interfaces for tabular data analysis. In *Proceedings of the 6th ACM Conference on Conversational User Interfaces (CUI '24)* (pp. 1–6). https://doi.org/10.1145/3640794.3665577
16. Yavorskyi, D. S., & Tkachenko, O. I. (2024). Some aspects of software mapping of tabular data of complex structure into an ontological model. In *Proceedings of the V International Scientific and Practical Conference on Management and Administration in the Conditions of Countering Hybrid Threats to National Security* (pp. 642–645). (in Ukrainian)
17. Tkachenko, O. I., & Tkachenko, O. A. (2017). Some aspects of situational-semantic modeling of complex objects, processes and systems. *Journal Water Transport, 1*(26), 129–133.
18. List, C. (2018). *Levels: Descriptive, explanatory, and ontological*. http://eprints.lse.ac.uk/87591/1/List_Levels_descriptive_2018.pdf
19. Edgington, T., Choi, B., Henson, K., Santaman, R., & Vinze, A. (2004). Adopting ontology to facilitate knowledge sharing. *Communications of the ACM, 47*(11), 85–90. https://doi.org/10.1145/1029496.1029499
20. Takeoka, K., Oyamada, M., Nakadai, S., & Okadome, T. (2019). Meimei: An efficient probabilistic approach for semantically annotating tables. In *Proceedings of the AAAI Conference on Artificial Intelligence, 33*(1), 281–288. https://doi.org/10.1609/aaai.v33i01.3301281
21. Kruit, B., Boncz, P., & Urbani, J. (2019). Extracting novel facts from tables for knowledge graph completion. In *Proceedings of the 18th International Semantic Web Conference (ISWC 2019)* (pp. 364–381). https://doi.org/10.1007/978-3-030-30793-6_21

22. Zhang, S., & Balog, K. (2020). Web table extraction, retrieval, and augmentation: A survey. *ACM Transactions on Intelligent Systems and Technology*, *11*(2), 1–35. https://doi.org/10.1145/3372117
23. W3C. (n.d.). *Generating RDF from tabular data on the web*. https://www.w3.org/TR/csv2rdf/
24. Iida, H., Thai, D., Manjunatha, V., & Iyyer, M. (2021). TABBIE: Pretrained representations of tabular data. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 3446–3456). https://doi.org/10.18653/v1/2021.naacl-main.270
25. Debellis, M. (2021). *A practical guide to building OWL ontologies using Protégé 5.5 and plugins*. https://www.researchgate.net/publication/351037551
26. Protégé. (n.d.). https://protege.stanford.edu
27. Fawei, B., Pan, J. Z., Kollingbaum, M., & others. (2019). Semi-automated ontology construction for legal question answering. *New Generation Computing*, *37*, 453–478. https://doi.org/10.1007/s00354-019-00070-2
28. *Data Integration: Using ETL, EAI, and EII Tools to Create an Integrated Enterprise.* (2007). http://www.tdwi.org/Publications/WhatWorks/display.aspx?id=7979
29. Majid, M., Hayat, M. F., Khan, M. F., & Ahmad, F. M. (2021). Ontology-based system for educational program counseling. *Intelligent Automation and Soft Computing*, *30*(1), 373–386. https://doi.org/10.32604/iasc.2021.017840
30. Sanfilippo, E. M. (2018). Feature-based product modelling: An ontological approach. *International Journal of Computer Integrated Manufacturing*, *31*(11), 1097–1110. https://doi.org/10.1080/0951192X.2018.1497814
31. Ziemer, J. (2025). *Configurable Ontology to Data Model Transformation (CODT)* [Patent].
32. Tijerino-Embley, D. W., Lonsdale, D. W., & Ding, Y. (2005). Towards ontology generation from tables. *World Wide Web*, *8*(3), 261–285. https://doi.org/10.1007/s11280-005-0360-8
33. Mansurova, M., Barakhnin, V., Ospan, A., & Titkov, R. (2023). Ontology-driven semantic analysis of tabular data: An iterative approach with advanced entity recognition. *Applied Sciences*, *13*(19), 10918. https://doi.org/10.3390/app131910918
34. Schreiber, G., Wielinga, B., & Jansweijer, W. (2003). *The KACTUS view on the "O" word*. https://www.researchgate.net/publication/2942056
35. Corea, C., Fellmann, M., & Delfmann, P. (2020). Ontology-based process modelling – Will we live to see it? https://doi.org/10.48550/arXiv.2107.06146
36. Nagypál, G. (n.d.). *Ontology development methodologies for ontology engineering*. https://link.springer.com/chapter/10.1007/3-540-70894-4_4
37. Fensel, D., van Harmelen, F., Klein, M., & Akkermans, H. (n.d.). *On-To-Knowledge: Ontology-based tools for knowledge management*. https://www.researchgate.net/publication/2412399
38. NeOn Project. (n.d.). https://oeg.fi.upm.es/index.php/en/completedprojects/8-neon/index.html
39. AIMS. (n.d.). *The NeOn project*. https://aims.fao.org/projects/neon
40. Khan, A. (n.d.). *5 tools to create your ontologies*. https://www.lettria.com/blogpost/5-tools-to-create-your-ontologies
41. Elixir Europe. (n.d.). *Ontology-related tools and services*. https://faircookbook.elixir-europe.org/content/recipes/interoperability/ontology-operations-tools.html
42. Cognitum. (n.d.). *Fluent Editor*. https://www.cognitum.eu/semantics/fluenteditor/
43. OntoStudio. (n.d.). https://www.w3.org/2001/sw/wiki/OntoStudio
44. Weiten, M. (n.d.). OntoSTUDIO® as an ontology engineering environment. https://link.springer.com/chapter/10.1007/978-3-540-88845-1_5
45. Sure-Vetter, Y., Angele, J., & Staab, S. (2003). OntoEdit: Multifaceted inferencing for ontology engineering. In *Lecture Notes in Computer Science* (pp. 128–152). https://doi.org/10.1007/978-3-540-39733-5_6