MODEL, DATA INTEGRATION ALGORITHMS OF INFORMATION SYSTEMS BASED ON ONTOLOGY

[1]RAISSA USKENBAYEVA, [2]TOLGANAY CHINIBAYEVA

[1]Professor International Information Technology University, Almaty 050040, Kazakhstan

[2] Senior Lecturer International Information Technology University, Almaty 050040, Kazakhstan

E-mail: [1]ruskenbayeva@iitu.kz,
*Corresponding author E-mail: [2]ctolganay@gmail.com

ABSTRACT

Today is difficult to imagine without integrated automated information systems (IAIS) that support various business processes. Obtaining aggregated information for the purpose of making management decisions depends on the effective interaction of information systems (IS) included in the structure of the IAIS. Business processes are often automated using software solutions of its own and third-party developers, regardless of their interconnectedness, which is especially true for universities. With constant changes in these business processes, IAIS developers are forced to adjust programs and data models, which leads to structural and semantic heterogeneity of information elements and, accordingly, the need to re-develop software data converters. The use of such solutions leads to complication and, consequently, a decrease in the reliability of the IAIS.

This article continues the series of publications [25-27], which includes the problem of building integrated automated information systems that support various business processes based on ontologies.

The purpose of the article is to develop a mathematical model, algorithm and software system for integrating IS data based on the application of the ontological approach.

To achieve this goal, the following tasks were set.

1. Build a mathematical model of data integration of information systems with heterogeneous ontological specifications.

2. To develop a computational method for evaluating the semantic proximity of concepts (elements) of heterogeneous ontologies.

Keywords: *Ontology description languages, data integration algorithms, genetic operator, lexical proximity, heterogeneous information systems.*

1 INTRODUCTION

One of the approaches to building an integrated automated information system is the integration of ready-made information systems.

The process of data integration in the construction of IS is understood as ensuring the interaction of individual subsystems. The result of integration is the achievement of unity and integrity within the system.

To improve the efficiency of integrated data processing, it is necessary to choose a method of integrating existing and ever-increasing information systems into a single information space. This need is especially acute when creating an integrated automated information system, which, on the one hand, provides users with access to relevant and consistent information, on the other hand, it is a necessary tool for the activities of university staff and student training. Analysis of correct and complete IS data affects the effectiveness of decisions made by the organization's management.

Most universities have a significant number of information systems that are responsible for certain business processes. IS is created at different times by different groups of developers using dissimilar technological solutions. Some of them may not be documented or supported by the developers. These systems are used, as a rule, by separate departments of the university. Only the users themselves (department employees) know about the structure and volume of stored information in the IS.

Data integration of many heterogeneous information systems of a university is the main problem in the construction of complex systems, the solution of which is achieved through functional, technical, and software compatibility. The main step in creating an IS should be documenting all developments using standards, which guarantees the creation of successful systems.

The main problems when integrating IS:

1. Providing structured source information that is stored in parts in different systems and can be duplicated. One approach to solving this problem is to bring all data to a single structure.

2. Providing uniform access to heterogeneous information systems, which were created on the basis of different technologies. One of the possible ways to solve this problem is to build a general integration architecture.

3. Ensuring information exchange of all systems in the information space of the organization.

4. The existence of various information models, as well as frequent changes in their structures, lead to the need to develop and improve methods and means of integrating heterogeneous information resources.

The number of physical databases and the implementation features of the DBMS that control them are not the main criteria for assessing the complexity of the integration process. The key concept is the subject area of IP. The subject area of IP covers a certain area of the organization. Thus, "the subject area includes a set of concepts that can be operated on" [1]. Relationships can exist between subject areas, for example, one subject area includes concepts from another. It is essential that there are fundamentally several subject areas related to the business processes of the university.

Each information system of the university covers its own subject area. This leads to structural and semantic heterogeneity, when “data from different sources can be presented and organized differently, or similar concepts can be interpreted differently in different data sources” [2].

The effect of heterogeneity is manifested with an increase in the number of heterogeneous heterogeneous information systems with different functionality [1].

IS heterogeneity has several aspects:

The heterogeneity of requirements. IS development is carried out on the basis of the requirements of the relevant subject areas and is changed in the process of maintenance in connection with changes in their features. In addition, it is obvious that the requirements for the system due to different subject areas can be contradictory, which makes it necessary to choose the most important one.

Difference in data models - "data in different ISs can be represented in different ways and in different data models" [2].

Syntactic heterogeneity - "data can have a different representation when transmitted over a communication channel in accordance with interaction protocols (for example, binary, text, XML, etc.)" [2].

Structural heterogeneity - in different information systems, data can be represented by different structures ("for example, a full name can be represented by one line or three lines") [2].

Semantic heterogeneity - "the same data can be represented in different systems of concepts, similar concepts can be interpreted in different ways in different IS" [2].

Technical heterogeneity - integrated information systems are implemented using various technical solutions, from different manufacturers, have different methods and protocols of interaction for accessing the system, etc. [2].

Heterogeneity of data access methods - in particular, "different purpose and expressiveness of query languages for retrieving data, different restrictions on the form of queries" [2].

In this work, the problem of integrating these information systems is considered as the problem of integrating their subject areas.

To solve the problem of semantic heterogeneity of information in the integration of information systems, one can use domain ontologies.

The domain ontology includes concepts and relationships between them. The use of a unified ontology of subject areas to a certain extent allows us to solve the problem of heterogeneity at the level of conceptual semantics. However, in the IS subsystems of a university, different requirements are imposed on subject areas, the depth and formality of their description, therefore, heterogeneous ontological descriptions of the subject area, presented in heterogeneous ontological models, can be used.

The heterogeneity of ontological specifications appears at the levels of model and conceptual semantics. Accordingly, there are problems of harmonizing ontological specifications. At the model level, differences are the factors that create heterogeneity:

- in the syntax of languages defining ontological models;

- in the expressive ability of models;
- in the semantics of primitives used in models.

At the ontological level, heterogeneity gives rise to differences:

- in the names of concepts and relations;
- in approaches to the definition of concepts;
- in breaking the subject area into concepts;
- in the coverage of the subject area;
- in terms of the subject area.

In order to correctly integrate heterogeneous information systems, it is necessary to find out the commonality and differences of the ontologies that underlie them, as well as agree on heterogeneous ontological specifications and then carry out information transformation. As a result, the joint work of heterogeneous information systems in the context of the problem domain at a semantically significant level is ensured.

An analysis of the state of research on ontology reconciliation shows that these topics have not yet been investigated deeply enough, mainly for particular cases. The existing methods are mostly informal and need to be improved when used in another organization.

When developing systems, the integration of ontologies is preferred to be avoided, despite the fact that this problem is relevant when using ontologies for the development of corporate information systems

1.2. Analysis of integrated automated information systems of universities

Today, universities use information technology as one of the powerful tools to improve the efficiency of work, teaching and research, as well as competitiveness. In a number of universities, work is underway to build an integrated automated information system to support educational, financial and management activities. Many IS subsystems are created on a variety of hardware and software in order to automate the activities of individual departments. From such systems, it is possible to obtain aggregated information in the form of a report only after the data converter programs have been revised for a specific task.

The main task of IS is to automate the key areas of the university [3, 4]: the educational process (support for conducting classes, control of knowledge, etc.); management of the educational process (distribution of the teaching load between departments, teachers; scheduling classes, etc.); management of research activities; financial planning; management accounting; administrative management (management of the organizational structure, personnel, decision support, etc.); information resource management (user access to data, systems, etc.). Figure 1 shows the information flows of the university.

"Communication is possible in three directions: a) vertical integration of information for strategic management, carried out on the basis of data obtained in the course of solving problems of operational management; b) horizontal integration of information - based on operations performed within the framework of solving tasks of one level of management; c) time integration of information - based on operations carried out with data related to different time periods. IS is also an important element in the implementation of feedback in the management scheme of a university. Based on information received from IS, decision makers, rector, vice rectors - have the ability to quickly assess the current situation, draw appropriate conclusions and form management decisions "[3].

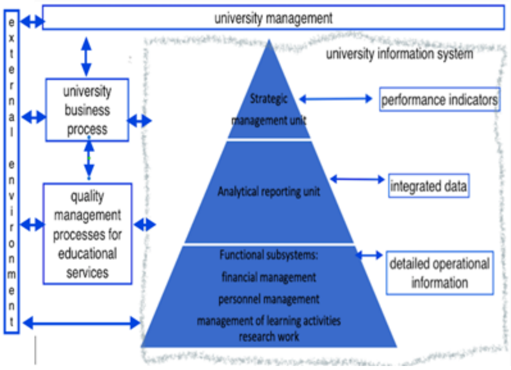
university management
external environment
university information system
university business process
quality management processes for educational services
Strategic management unit
performance indicators
Analytical reporting unit
integrated data
Functional subsystems:
financial management
personnel management
management of learning activities
research work
detailed operational information

Figure 1: Place of IS in the university management scheme

The IS of the university can be built on the basis of a unified technology (DBMS), which does not require data integration. But the disadvantages are:

1) limited implementation of various functional requirements (there may be a need for functionality that cannot be implemented in the development technology used);

2) the impossibility of linking into a single whole network, organizational infrastructure;

3) limited use - by individual divisions or by users.

An analysis of the IS of universities based on various technological solutions [5] showed that in most cases the integration of their own developments is carried out, which are used for the tasks of managing the educational process, scientific activities, and third-party systems for administrative management tasks. IS, developed on the basis of the concept of integration, cover various areas of the university and provide access to data for multiple users, usually they are "the environment of the entire university" [5].

However, in universities, most information systems:

- do not provide comprehensive support for making management decisions (for example, support for quality management processes);
- are fragmented;
- poorly adapted to adapt to changing functional requirements.

As the analysis showed, the university, due to the large number of ISs that automate various business processes, has not solved the problem of integrating heterogeneous information resources.

1.3 Basic approaches to the integration of information systems

Data integration is one of the highest priority tasks of building IS. The problem of data integration arises when using information systems from different developers or with different database management systems (DBMS), as well as where it is necessary to get access to aggregated information.

Before data integration, it is necessary to "identify and catalog the data, and build a data model" [5]. In practice, the most suitable implementation technology (EDI, DCOM, OLAP, GIS, XML, Web services) is used to solve the data integration problem [5, 6].

There are several methods of data integration of information systems.

The method of data dissemination is to transfer information from one IS to another after the occurrence of certain events. A distinctive feature of this method is the operational data exchange. Data is transferred both synchronously and asynchronously. The disadvantages include the impossibility of executing general analytical queries, since it may be necessary to use a temporary storage-analyzer, which is not provided for in this approach.

When using the consolidation approach, data is extracted from multiple information systems and placed in one data warehouse. The repository seeding process is unidirectional and is divided into three phases - extract, transform, and load. There are several modifications of this approach, which can be attributed to the following categories: structure transfer and integration [7, 8, 9].

Migration also implies merging data structures. The integration process consists of combining the data model, metadata and the data itself in a new IS.

In order to minimize costs, the developers of organizations use the transfer of structures. This allows you to reduce the number of servers used and, accordingly, the costs of their maintenance. It should be noted that transferring only structures is not part of the data integration process.

The main stages in consolidation include:

• definition of a single standard for IS and transition to it (definition of data warehouses and data marts);

• creation of storage and data marts. This repository collects all data from source information systems and integrates them logically using identification keys and common measurements. This reduces the number of agent programs to retrieve data;

• synchronization. In order to store general information about objects, a centralized operational data warehouse is used.

Along with the data federation method, on-demand data integration technology is used. In this case, a single virtual information space is formed and integration takes place in real time [10, 11]. "The integration server accepts XQuery queries, parses these queries, separating individual queries to different data sources, optimizes them. Thus, if a query contains a call to several data sources, then it is divided into several separately executed subqueries. To obtain a resultant response, the results of the subqueries sewn together "[4]. The disadvantages of the federalization method include low performance, which does not allow the use of servers in many tasks of the IS of the university.

"With logical integration, based on existing descriptions, there is no need to generate specialized XQuery queries. One of the problems of the data integration server is the impossibility of describing the relationships between semantic objects, that is, context-sensitive relationships that are present in complex systems" [4].

In [4], data integration is considered "on the fly"; in real time, and it is an urgent task of building IS in the university.

This approach solves the problem of not only the joint functioning of IS subsystems, but also the problem of their maintenance, since all changes in the integrated subsystems immediately become available to all users of the integration system and there are no special requirements for the hardware [4].

In the paper, the integration of information systems is understood as the process of establishing the mapping of heterogeneous ontologies of IS to ensure the joint operation of these systems.

Companies such as Oracle, Microsoft, SAP, Business Objects, Sybase, SAS Institute, Cognos stand out among the leaders in the data integration software market.

IBM and Informatica are both involved in data integration technologies. The data integration product group provides data quality, support for unstructured sources of metadata management. A distinctive feature of this architecture is "a metadata management environment and powerful mechanisms for parallel processing of large amounts of data". It supports role-based interfaces that enable users to work with data.

PowerCenter 8.5 is designed as a unified data integration platform, has a centralized metadata infrastructure, and integration functions are implemented as services. The system supports unstructured data access, batch data delivery, real-time delivery, and retrieval of modified data only. PowerCenter AdvancedEdition allows you to extract metadata from a variety of sources and also provides metadata analysis tools.

One of the popular integration products is Service Oriented Architecture (SOA) (provides application and data integration). The main idea behind this approach is to view the most important business functions of an organization as a collection of services. The data is in the source systems, and even the location of the data is unknown, each data source is associated with a specific service. When requested from the user, services are called. One of the disadvantages of this approach is the limited number of requests to retrieve information [12, 13].

The SOA architecture is based on the Web-Service technology [4]. Web services exchange messages using specific protocols: WSDL (Web Service Request Description Language), SOAP (Structured Messaging Protocol), and XML (Extensible Markup Language) [4]. The main advantages of this technology are: independence from the IS development environment and ease of creating web services. There are also a number of security and performance disadvantages [29, 4].

The Enterprise Service Bus (ESB) approach is based on the use of asynchronous messaging between information systems through a single point according to the SOA principle [4]. Using this approach, you can provide access to data from third-party information systems. Based on the work [4], the disadvantage of this approach is that the integration of a new information system requires "the involvement of developers who implement the necessary functionality in information systems, ensuring the generation and processing of messages" [4].

Thus, there are many different data integration solutions from different software vendors. It should be noted that actively developing these solutions make it possible to solve almost any integration problem by modifying it for specific information systems.

Therefore, information systems cannot be limited only to the use of one of these solutions, but it is necessary to provide support for other technologies that ensure data integration.

Constant changes in business processes and infrastructure of an organization require modification of information systems subsystems and data models, which leads to re-development of converter applications for data integration.

The analysis showed that the existing integration technologies ensure data integration at the physical, logical and semantic levels. Integration at the physical level seems to be the simplest task, since data from heterogeneous sources are converted into the required universal format. When integrating data at the logical level, the local data model is mapped to the global one. This generates a number of conflicts, in particular: the use of different terms to designate the same concepts; various kinds of semantic conflicts. Given the complexity of the task at hand, to solve the problem of heterogeneity, it is necessary to use a data integration approach based on semantics.

1.4 Using the ontological approach as a basis for the integration of information systems

Today, a significant amount of knowledge is accumulating in heterogeneous information systems. And when integrating such systems, the problem of systematization and structural representation of knowledge about different subject areas arises. To solve this problem, ontological models can be used in order to obtain a formal specification of conceptualization. Ontological specification includes a combination of a formalized description of knowledge in the form of axioms and an informal description.

As noted in [15, 16], the ontology-based approach is used in various fields from knowledge representation to information integration. Ontology is used in knowledge management systems in order to formally describe the modeled part of the world in the form of a dictionary shared by specialists in the selected subject area. Based on this common vocabulary, various sources of knowledge can be integrated. Thus, using a common vocabulary, it is possible to understand and compare various information systems.

Ontology can be used at the stages of development and operation of an information system.

Ontology is defined as a specification of the conceptualization of the subject area and as a means of representing the semantics of information units [7]. As noted in [1], "the domain ontology can be used to describe information objects, their properties and relationships." Information about these objects can be stored in different sources, and to see the complete picture, it is necessary to build an information model [2].

In the works of scientists [1, 4] in knowledge management systems ontology is the basis for the formation of an organization's knowledge base. In work, an ontology is used to create an information and educational environment. In these cases, we are talking about the organization of intelligent storage of unstructured data. At the same time, such a data warehouse is not connected with other subsystems of information systems.

At present, automated data processing technologies cover the level of their semantics. With an increase in the amount of processed information from heterogeneous data sources, the problem of heterogeneity of ontological specifications arises. Therefore, there may be problems of harmonizing ontological models. The article proposes a method for integrating heterogeneous ontologies of subject areas.

Ontologies are classified as follows:

Top level. It contains general knowledge for several subject areas. It can describe the most general concepts such as space, time, event, object, action, etc.

Domain-oriented. The purpose is similar to the top-level ontology, but the area of interest is limited to one subject area (for example, the educational process). At the same time, it can use the specialization of terms that are located in the top-level ontology. It is used by subject matter experts to annotate information.

Task-oriented. These are ontologies used by specific application programs and containing terms that are used in the development of information systems that perform specific tasks.

Quite often, within the framework of one organization, ontologies of subject areas are developed and combined by different uncoordinated groups of experts. This leads to the fact that several semantically heterogeneous or independently developed from each other ontologies describe one subject area. Subsystems in information systems have their own particular ontologies and organizational differences. In such conditions, the tasks of displaying and integrating ontologies inevitably arise. In the general case, the integration of two ontologies will mean the process of creating a new ontology based on finding the similarity of their elements, taking into account the semantic features.

1.4.1 Methodology for building ontologies

The methodology for constructing an ontology involves solving the following problems [15]:

• define the purpose and scope;

• build an ontology using a specialized knowledge representation language;

• achieve a common understanding of the structure of information;

• ensure the use of knowledge in the subject area;

When creating an ontology, it is necessary:

1. Conduct an ontological analysis. A glossary of terms is compiled, which includes a description of the characteristics of objects and processes included in the information system. The logical relationships between the concepts of the subject area are also described;

2. to highlight concepts - basic concepts;

3. determine the number of levels of abstraction;

4. distribute concepts by levels;

5. Build connections between concepts - define relationships and relationships with basic concepts;

6. to consult with various experts to eliminate contradictions and inaccuracies.

The ontology contains a set of terms and rules by which one can build reliable statements about the state of the information system under consideration.

The article uses the IDEF5 methodology to model and visualize ontologies. It developed "special languages that are used to represent information about the ontology in a transparent graphical form" [19].

When constructing an ontology, the following must be done:

1) create a dictionary of terms;

2) describe the rules and restrictions by which reliable statements can be formed;

3) based on the statements, build a model that allows you to form the necessary additional statements.

There are four types of diagrams in the IDEF5 methodology:

• "classification diagram (Classification Schematics) - intended for the logical systematization of knowledge";

• "Composition Schematics - intended for graphical presentation of the composition of ontology concepts";

• "Relation Schematics - for visualization and study of relationships between concepts in ontology";

• "object state diagram (Object State Schematics) - a tool for documenting processes in terms of changing the state of an object".

In the IDEF5 methodology, knowledge in the form of a set of concepts, attributes and relationships is used to build a conceptual model. Thus, using the IDEF5 methodology, it is possible to visually represent the state of objects throughout the entire process and effectively develop and study an ontology.

The structure and properties of the information system of the university can be analyzed using a dictionary of terms in order to describe the characteristics of objects and processes related to the system.

1.4.2 Formal Ontology Model

Ontology is a formal explicit description of concepts (concepts) in the considered domain, properties and attributes of each concept (slots), and restrictions imposed on slots (facets). Slots are sometimes referred to as roles [29]. "Ontology together with a set of individual instances of concepts forms a knowledge base. In reality, it is difficult to distinguish between an ontology and a knowledge base" [3].

In [1], the formal model of the ontology is presented in the form of a "triplet of finite sets" O = <T, R, F>, where:

T - terms of the subject area described by the ontology O;

R - relations between terms of a given subject area;

F - interpretation functions defined on terms and/or relations ontologies O [1].

Relationships mean the type of interaction between concepts

("Part-whole", "is a subclass", "has an effect", "similar to", etc.). The axioms [1] are used to model statements.

To describe complex systems, such as the IS of an university, such a concept as an extensible ontology model is introduced.

1.4.3 Methods for assessing semantic proximity in ontologies

The ontological approach provides a new level in solving information integration problems.

To ensure the semantically correct interconnection of heterogeneous information systems, it is necessary to compare the ontologies that underlie them and find out their commonality and differences. This problem is solved by using methods for assessing the semantic proximity of ontology concepts.

Many well-known methods for finding a measure of proximity between ontology concepts are based on Tversky's set-theoretic approach, based on comparing the properties of concepts.

In [22, 21, 6, 20], the structure of paths between concepts is considered, namely, the length of the shortest path is determined as the number of concepts in the hierarchy between the two considered concepts in the ontology, "the shorter the path length, the closer they are" [22].

In [7], proximity is estimated as a semantic distance, it is inversely proportional to the semantic proximity of concepts.

In [8], the measure is based on the frequency of occurrence of the concept and its subclasses, and is interpreted as a probability value.

The disadvantage of the methods described above for calculating proximity measures based on ontological structures is their symmetry. According to expert estimates, the measure of proximity is not always symmetrical in most cases.

In [9], an asymmetric measure of semantic proximity is proposed, which takes into account the directions of passage along the edges. Here, the concept child is semantically closer to the concept parent than the concept parent is to the concept child.

In [10], a calculation method is described, the essence of which is that the proximity of two concepts depends on the proximity of concepts with which there are hierarchical relationships, and is calculated recursively.

Hybrid measures that combine several approaches seem to be the most promising.

The hybrid measure proposed in consists of three parts - taxonomic, relational, and attributive.

Difficulties in comparing different ontologies of subject areas lie in the difference in the names of concepts and relations, as well as in the approaches to the definition of concepts. When mapping two ontologies, a search is performed for each concept of one ontology of a similar concept of another ontology.

In [24], to establish a mapping of two ontologies, the concept of bridges is used — vertices in a taxonomy that correspond to equivalent concepts. The closest common parent of the compared concepts is taken into account here.

In [11], a method for calculating a measure is proposed, taking into account the lexical proximity of concepts, properties, domains and ranges of relations (ranges of values of the arguments of relations), parent / child concepts.

The works [10, 14] consider the verbal and conceptual levels, where the lexicons are compared and the taxonomies of the concepts of two ontologies are compared.

The main disadvantage of most methods for determining semantic proximity is the need to involve an expert to confirm the correctness of detecting similarities and differences in semantic concepts.

1.4.4 Ontology description languages

In order to implement an ontology, it is necessary to choose an Ontology specification language that has sufficient expressive power. Such a language makes it possible to indicate the machine-interpreted semantics of systems and brings it closer to the real world, which significantly increases the expressive capabilities of conceptual modeling.

Ontology specification languages are divided into simple ones, into languages based on descriptive logic and frames (OKBC, OCML, Flogic), as well as on Web standards (XOL, RDF (S), DAML, OIL, OWL, SHOE, UPML) [1].

Traditional languages and Web-languages of ontology specification (Ontolingua, CycL) are distinguished by expressive capabilities of the domain description and some inference mechanisms. They include constructions for multiple hierarchies of concepts, inference rules, axioms, as well as the ability to record ontologies and relationships between them.

Language RDF (Resource Description Framework). This standard for describing metadata is based on XML syntax. "RDF uses a data model - object, attribute, value". RDF allows you to describe many objects of information systems in the form of a directed graph. RDF dictionaries use a basic structure to describe concepts and the types of relationships between them. This standard can be expanded. For example, you can define the structure of a source description using concepts such as concepts, properties, types, collections. There are several software products that allow you to describe RDF triplets for various types of information systems. RDF Schema (RDFS) is a standard for describing domain models using resources, properties and their values. Thesauri can be described by means of RDFS. One of the disadvantages of RDFS is the impossibility of specifying axioms and inference rules based on them.

DAML + OIL [22] is a semantic language for creating ontologies that contains a rich set of constructs (headers, concept elements, property elements, instances).

OWL (Web Ontology Language) is "a language for describing ontologies based on the XML, RDF, RDFS and DAML + OIL standards". The language is based on the data model "object - property", which allows you to describe classes and the relationship between them. The formal basis of this language is descriptive logic, which allows you to significantly expand the expressive capabilities of the system.

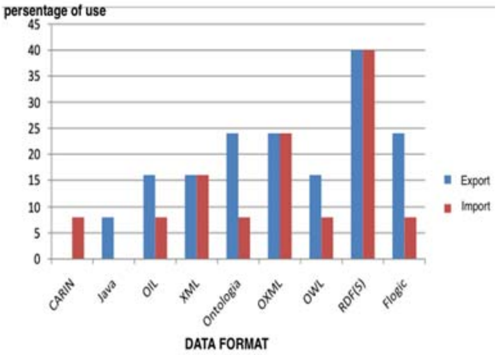

Figure 2: Diagram of the use of various formats for describing ontologies when importing and exporting data

To build ontologies, many ontology editors have been developed that support various languages and formats for representing ontologies. According to the study [1], the results of which are presented in Figure 2, the most commonly used format for representing ontologies is RDF (S). This standard has a convenient

Percentage of use for perception by the form of data representation in the form of a directed graph.

1.4.5 Tools for ontology processing

It is possible to speak about the task of integrating ontologies in order to harmonize ontological concepts only when heterogeneous ontologies reflect either different points of view on the same subject area, or on overlapping subject areas.

One of the advantages of using ontology for the integration of information systems is the availability of software that supports ontological analysis. There are a number of tools (Ontolingua, OntoEdit, OilEd, WebOnto, ODE) that support editing, documenting, visualizing, importing and exporting ontologies, as well as merging and comparison. Let's consider the most famous of them.

Protégé is a freeware editor for designing domain ontologies. An editor to build ontologies using the OWL language.

Such tools are used both for design and for ontology analysis, performing typical operations, for example:

- alignment - finding and establishing correspondences in both ontologies;
- mapping - finding semantic dependencies between elements of different ontologies;

• union - creation of the resulting ontology for the other two.

PROMPT is an extension of the Protégé system and serves to unify and group ontologies. PROMPT displays to the user a list of operations for combining two ontologies, as well as a list of conflicts and their possible solutions. "The user selects the required action, the list is formed again until a new ontology is ready".

Chimaera is "a program for integrating ontologies based on the Ontolingua editor". It provides the user with the result of the analysis of the union of ontologies, pointing out problem areas, but does not take any action on its own. Only concept names and their taxonomy are checked.

OntoMerge provides tools for translating data into a generic representation in a special language. Then the axioms of connection between the concepts of two ontologies are determined.

OntoMorph offers a set of actions that can be applied to an ontology.

OBSERVER allows you to find synonyms in the original ontologies and provides the user with information about the mapping of ontologies for formulating queries in terms of their own ontology.

ONION is a tool for combining ontologies using ontology algebra.

Developments in this direction for the mapping of ontological models are mainly related to specific models or their concepts. There are few works on methods of displaying arbitrary information models. Some of them are based on methods for constructing functions for displaying model elements. Noteworthy is the research [18] that is devoted to mapping models based on specifications.

The solution of the ontology mapping problem is reduced to the construction of an interface to the system that uses the most general means of ontology specification. Thus, the ontological model OKBC (Open Knowledge Base Connectivity) uses the frame model as a basis and serves as an exchange language for ontologies. The programming interface is considered as a key tool for the architecture of a distributed ontology repository. OKBC is part of the Ontolingua server. Constructs consist of classes, frames, slots and some relationship properties.

The existing methods of displaying ontological models do not take into account their specificity. Therefore, in this case, one can focus on research on the mapping of information models that are not related to ontology. The specificity of ontology mapping is associated with the types of entities and relationships used in many ontological models.

Using the existing methods, it is impossible to integrate ontologies created by different working groups without the participation of an expert.

2 USING RDF TO BUILD INFORMATION SYSTEMS MODELS

RDF is a data presentation model. RDF uses elements from SGML and XML. "Entities are described by specifying their properties and the values of these properties" [1]. RDF resource claims are:

• the described entity - the subject;
• property is a predicate;
• property value - object.

URI (Uniform Resource Identifier) and URI reference (URI with fragment identifier) are used to designate subjects, objects and properties [1].

The structure of an RDF specification consists of a description of all information objects. The latter consists of a link to the described information object (via URI) and descriptions of properties (Name; value; link to another information object).

To express semantics, you need to create a dictionary of terms. For this purpose, you can use the language RDFS, which is an extension of RDF. It contains tools for defining classes, properties and rules.

Benefits of using RDF:

• RDF is web-oriented and is good at
• scalable;
• RDF - ontologies are published by any user on the web in order to expand existing concepts (relationships) with new concepts, if required;
• URIs are used as global identifiers for all concepts, making it easier to manage the global URI namespace through the use of the Domain Name System (DNS).

Thus, the description of an ontology using OWL and RDF/RDFS technology allows better expressing the semantics of all entities and their internal and external relationships. As a result, they can be used for the effective functioning of the developed ontology integration algorithms.

2.1 Mathematical model of data integration of information systems

An integration algorithm is proposed based on the results of comparing concepts, their attributes and relationships between concepts [13]. In addition, a generalization of the problem of combining ontologies to the procedure of integrating ontologies describing different subject areas of the same domain of subject areas is proposed. The task of integrating information systems is reduced to the task of constructing mappings and integrating ontologies, and then establishing interconnections between the schemes of integrated information systems, i.e. preserving the correspondence of a set of ontologies of information systems to a given set of semantic dependencies, allowing the establishment of interaction between information systems. RDF statements are used to describe the ontological specifications of information systems to be integrated.

As a rule, the object schema of information systems includes elements that correspond to the entities of different subject areas, each object is characterized by the values of a set of attributes and is represented as a set of ordered pairs of the form

$$u = < a_i, d_i >, \qquad (1)$$

where a_i is an attribute of the object; d_i - the value of the attribute $i \in [1 \ldots n]$; n - number of attributes.
The basic concept of the proposed model is concept C (class of objects). Each concept of an information system ontology defined as a unit of knowledge and identified by name and characterized by type.
Therefore, we define the concept as

$$C_i = (Name_i, type_i), \qquad (2)$$

where $Name_i$ is a unique name (identifier) of the $i -$ concept; $type_i$ - the type of the i-concept (abstract, representable, or composite). An abstract type is a list, an array, etc.

The representable type is numbers, strings, images, etc. Composite type is an aggregation of heterogeneous or homogeneous structures (concept, attribute, relation). Below is an example that defines concepts with abstract types

(Premises, Streaming Lecture Audience, Room Capacity).
{Ontology Auditor Fund; in: module;
kind: ontology; type:
{Premises;
in: type, concept;
Capacity of the Premises: Capacity of the Premises; metaslot
inverse: Capacity of the room.From the room.
end
},
{Stream Lecture Audience; in: type, concept;
supertype: Premise; Area of the room:
{in: predicate, invariant;
{predicative; {
all a / Stream Lecture Audience (a. Room Capacity> 50)
}}}},
{Capacity of the Premises; in: type, concept; Room: Premises;
metaslot
inverse: Room.CapacityPlace;
end
};
}

Let's set the following set of concepts $C = \{C_1 | i = 1,2, \ldots, n\}$ and a set of relations between concepts

$$R = R_1, R_2, R_3 \ , \qquad (3)$$

where R_1 is an inheritance relation (class-subclass relationship), $R_{1,}C_1, C_2$
where C_1 is a superclass of the C_2 concept;
R_2 - aggregation relation (part-whole relationship), $R_2(C_1, A)$ attributes of the C_1 concept are included in the set of attributes of all A concepts.
R_3 is an association relation (semantic relations) with the property of transitivity.
Formally, we represent the ontology of an information system in the following form:

$$O \ =< C, A, L, P_A, P_C, R >, \qquad (4)$$

where $C = \ C_i \ i = 1,2, \ldots, n$ – many concepts;
$A = \{a_{ij} | ij = 1,2, \ldots, j\}$ – many attributes of concepts;
$L = \{l_{ik} | ik = 1,2, \ldots, k\}$ – vocabulary that defines the professional terms of the organization;
$P_A: C \to 2^A$– a mapping that defines a set of attributes for each concept;
$P_C: C \to 2^L$ – functions and concept interpretation, maps to a concept a set of vocabulary terms L;
R – many relationships between concepts. An information system using the O ontology is presented in the form

$$U^0 =< O, U, P_U, P_R >, \qquad (5)$$

where $U = \{u_1, u_2, \ldots, u_n\}$ – set of elements of the IS object scheme;

$P_U: U \to C$ – mapping that associates an object schema element with its concept;

$P_R: U \times U \to R$ – a mapping that associates relations between the elements of the object schema in the ontology, and for any element $u \in U$ the following condition is satisfied: the set of attributes of the object schema element u corresponds to the attributes of its concept, i.e.$\{a: < a, d > \in u\} = P_C \ (P_U(u))$

We denote by H^O the set of heterogeneous information systems based on the ontology O.

We denote the change in the information system as a map

$$F: H^O \to H^O \qquad (6)$$

Ontology change

$$U^O = \left\{U_1^{O^1}, U_2^{O^2}, \dots, U_N^{O^N}\right\}, \qquad (7)$$

where $U^O = \left\{U_1^{O^1}, U_2^{O^2}, \dots, U_N^{O^N}\right\}$,

$U_1^{O^1} =< O_i, U_i, P_{U_i}, P_{R_i} >$ and $O_i =< C_i, A_i, L_i, P_{C_i}, P_{A_i}, R_i >$ and introduce the notation: $\bar{C} = U_{1\le i\le N} C_i$, $\bar{R} = U_{1\le i\le N} R_i$, $\bar{A} = U_{1\le i\le N} A_i$, $\bar{L} = U_{1\le i\le N} L_i$, $\bar{U} = U_{1\le i\le N} U_i$

Different ontologies of information systems included in O may have overlapping sets of attributes, relationships, and concepts. On the basis of several initial ontologies, the resulting ontology is built while maintaining the original specifications in such a way that it includes all possible relationships between concepts and does not contain equivalent (duplicate) concepts. For this, it is necessary that the mappings P_U, P_C, P_A, P_R on the same ontology concepts coincide.

The resulting ontology determines the correspondence of concepts and the rules for their interpretation between information systems, which allows them to successfully establish their interaction.

An information system $U' =< \bar{O}, \bar{U}, \overline{P_U}, \overline{P_R}, >$ is called integrated on the set of information systems U^O if $U^O = \left\{U_1^{O^1}, U_2^{O^2}, \dots, U_N^{O^N}\right\}$, consistently, i.e. exist $\overline{P_U}: \bar{U} \to \bar{C}$, $\overline{P_A}: \bar{C} \to 2^{\bar{A}}$, $\overline{P_C}: \bar{C} \to \bar{L}$, $\overline{P_R}: \bar{U} \times \bar{U} \to \bar{R}$ which is extension of the corresponding mappings P_{C_i}, $P_{A_i}, P_{R_i}, P_{U_i} (1 \le i \le N)$.

2.2 Semantic dependencies

The construction of the mapping of the ontology O_1 to the ontology O_2 consists in finding for each concept of the ontology O_1 a similar concept of the ontology O_2.

As a rule, information systems should not only correspond to a certain structural scheme, but also satisfy more stringent requirements that are imposed by various semantic dependencies. Such dependencies determine the permissible states of the information system and are used to consistently change data in information systems. To implement a coordinated change of data into information systems in the context of the problem domain, it is necessary to find out the commonality and differences of information systems ontologies, to agree on ontological specifications. For this, semantic proximity is determined and semantic dependencies are established between the elements of ontologies (concepts). Thus, the goal of integration is to preserve the correspondence of the set of ontologies of information systems to a given set of semantic dependencies.

The semantic dependence defined on the ontology O is taken as a z-predicate defined on $\bar{O}$.

If there is a semantic dependence z in the ontology O, then we will write $z(O)$.

The set of semantic dependencies $Z = \{z^1, z^2, z^3, z^4, z^5\}$ has the consistency property, if $\exists \; \exists \; z_i(O) \forall i (1 \le i \le 5)$.

Consider 2 ontologies $O_1$and O_2:

$O_1 =< C^1, A^1, L^1, P_A^1, P_C^1, R_1 > -$ ontology of one information system;

$O_2 =< C^2, A^2, L^2, P_A^2, P_C^2, R_2 >$ − ontology of other information systems;

$n -$ number of concepts in ontology O_1;

$m -$ number of concepts in ontology O_2.

In practice, the dependence between ontologies must be reduced to the dependencies between the concepts that they include. They were reviewed, analyzed and assigned to the following 5 groups:

1. Equivalence $z^1: map(C_1) = C_{2,}, if\ S(C_1, C_2) \ge b$, where b is the threshold value of the semantic proximity measure $S(C_1, C_2)$, at which the mapping of the concept C_1 to the ontology O_2 is constructed.

The coincidence of all attributes of the concept C_i^1 of one ontology O_1 with all attributes of the concept C_j^2 of another ontology O_2 (element-wise equality of the sets A^1, A^2) means equality of the content of the two concepts.

2. Generalization $z^2: map(C_1) = C_2, C_2 = \{C_{2i}\}, if\ q < S(C_1, C_2) < b$, where b is the threshold value of the semantic proximity measure $S(C_1, C_2)$ at which the concept map is constructed C_1 to ontology

O_2; q - similarity threshold for establishing the lack of equivalence of concepts.

3. Refinement $z^3: map(C_1) = C_2, C_1 = \{C_{1i}\}, if\ q < S(C_{1i}, C_2) < b$, where b is the threshold value of the semantic proximity measure $S(C_1, C_2)$, at which the concept map is constructed C_1 to ontology O_2; q - similarity threshold for establishing the lack of equivalence of concepts.

4. Partial equivalence $z^4: map(C_1) = C_2$, $if\ q < S(C_1, C_2) < b$, where b is the threshold value of the semantic proximity measure $S(C_1, C_2)$, at which the mapping of the concept C_1 to the ontology O_2 is constructed; q - similarity threshold for establishing the lack of equivalence of concepts.

The intersection of the sets of attributes of the concepts C_2 and $C_1(A^2 \cap A^1 \neq \emptyset)$indicates the presence of common attributes. This means that there is some concept C, which is a superclass for concepts C_2 and C_1, and the concepts themselves belong to the same level of the hierarchy.

5. Difference $z^5: map(C_1) = \emptyset, \exists C_1, \forall C_2 \in O_2, S(C_1, C_2) < q$ where q is the similarity threshold for establishing the absence of concept equivalence.

The model of a data integration system based on ontologies is represented as a tuple

$$S \ =< O, U^O, Z, F, map >, \qquad (8)$$

where $O \ =< C, A, L, P_A, P_C, R >$ - ontology information systems;

U^O- information system with ontology O;

$Z = \{z^1, z^2, z^3, z^4, z^5\}$ – many semantic dependencies;

$F: H^O \to H^O$ – a mapping such that $\forall U^O \in H^O, \forall z \ \in \ Z$ completed $z(F(U^O))$;

$map: O_i \ \to \ O_j$ – ontology mapping

2.3 Method for evaluating semantic proximity between information systems ontologies

The ontologies of integrable information systems are initially not connected in any way; therefore, it is necessary to find semantically similar elements of ontologies.

The construction of a mathematical model for the integration of information systems, taking into account the comparison of their ontological specifications, creates an opportunity to measure the proximity (similarity) of ontological concepts.

In solving problems of displaying and integrating ontologies, information retrieval and building queries, entering new documents, an important role is played by the assessment of the semantic proximity of concepts and instances. Initially, the assessment of semantic proximity was based on the statement: the more information separates two concepts, the closer they are (geometric approach). But then a more objective set-theoretic approach was proposed by Tversky. Its idea is that in order to assess the semantic proximity, it is necessary to take into account not only the general properties of objects, but also their various properties.

For a numerical assessment of the semantic proximity of ontology concepts, an approach based on the research results of A.F. Tuzovsky and professor at the University of Mannheim A. Maedche. In the proposed method, the proximity measure consists of three parts. Attributive measure (comparison of concept attributes and attribute values), taxonomic measure (determination of the degree of similarity of ontology concepts based on their relative position, the length of the shortest path is calculated as the number of concepts in the hierarchy between the two considered concepts in the ontology, the shorter the path length, the closer they are) and a relational measure (takes into account relationships with other concepts).

This method has been adapted to calculate the semantic proximity of two heterogeneous ontologies. The modification of this method consists in replacing the taxonomic component with lexical correspondence (lexical proximity is based on the distance between two concepts of ontologies (the number of characters for transforming one lexeme into another)), as well as in the method of finding the attributive component and the use of a genetic algorithm for finding weight coefficients. In this case, the definition of the lexical component is calculated as the ratio of the intersection of sets of words (synonyms) in terms of their union. The main advantages of the proposed approach are in finding key concepts, eliminating the subjectivity of their descriptions and dependence on the points of view of ontology developers.

Let us define $S^T(c_i, c_j), S^R(c_i, c_j)$, $S^A(c_i, c_j)$, respectively, as a measure of the proximity of two concepts based on their position, based on the comparison of their relations, based on the comparison of attributes and values attributes of concepts.

The weighting factors t, r, a allow you to regulate the process of calculating the semantic proximity of two concepts.

To assess the lexical proximity of two concepts $S^T(c_i, c_j)$, the sets of terms of the concepts $PL_p(c_i)$ and $PL_p(c_j)$ are compared, common and different elements are found:

$$S^T(c_i, c_j) = \begin{cases} 1, if\ c_i = c_j \\ \frac{|PL_p(c_i) \cap PL_p(c_j)|}{|PL_p(c_i) \cup PL_p(c_j)|} if\ c_i \neq c_j \end{cases} \qquad (9)$$

where $PL_p(c_i) = \{L_i \in L | P_C(c_i) = L_i\}$ − many lexical terms of the concept c_i.

Below is an example of the many terms of the *Audience* concept.

{ The audience;
in: metaclass;
Term_section:
{Room;
Premises;
Conference hall;
Scene;
Tribune;
Board
}
}.

To assess relational proximity, it is assumed that if two concepts have the same relationship R_1, R_2, R_3 with the third concept, then they are more similar than two concepts that have different relationships.

Let's pretend that $C_r(c_i) = \{c_j \in C | R_1(c_1, c_2) \vee R_2(c_{i,} c_j) \vee R_3(c_{i,} c_j) \vee c_j = c_i\}$ − set containing concepts that have relationships R_1, R_2, R_3. Let us define the concept associativity relation as

$$R_A(c_j) = \{c_i : c_i \in C_r(c_j)\} \qquad (10)$$

Let's calculate the sum of the values of the lexical measure of proximity for concepts from the set $R_A(c_j)$ and $R_A(c_i)$.

$$S_{R_A}\left(R_A(c_i), R_A(c_j)\right) = \sum c_i \in R_A(c_i), c_j \in R_A(c_j) S^T(c_i, c_j) \ (11)$$

The relational proximity measure $S^R(c_i, c_j)$ allows you to assess the similarity of two concepts based on the similarity of concepts from the set $C(c_i)$.

$$S^R(c_i, c_j) = \begin{cases} 1, if\ c_i = c_j \\ \frac{S_{R_A}\left(R_A(c_i), R_A(c_j)\right)}{R_A(c_j) \cup R_A(c_i)} if\ c_i \neq c_j \end{cases} \qquad (12)$$

Let's compare the attributes of the two concepts.
Let's set a set of attributes belonging to the concept c_i.
$A^{c_i} = \{A_k^{c_i}, k \in [1 \dots n_1]\}$ where n_1- number of concept attributes c_i.
$A^{c_j} = \{A_k^{c_j}, k \in [1 \dots n_2]\}$ where n_2- number of concept attributes c_j.
The attributive measure of proximity $S^A(c_i, c_j)$ of concepts c$_i$ and c$_j$ is determined by the correspondence of their common attributes $A^{c_i} \cap A^{c_j}$.
The attributive measure of proximity $S^A(c_i, c_j)$ satisfies the axioms of independence and decidability and is defined by the formula

$$S^A(c_i, c_j) = \frac{|A^{c_i} \cap A^{c_j}|}{|A^{c_i} \cap A^{c_j}|} \qquad (13)$$

where A^{c_i} −many concept attributes c$_i$;
A^{c_j} − many concept attributes c$_i$.

The measure of proximity $S(c_i, c_j)$ of concepts c$_i$ ontology O and c$_j$ ontology O' is defined as

$$S(c_i, c_j) = (t \cdot S^T(c_i, c_j) + r \cdot S^R(c_i, c_j) + a \cdot S^A(c_i, c_j)), (14)$$

where t, r, a − coefficients that determine the importance of proximity measures $S^T(c_i, c_j), S^R(c_i, c_j),\ S^A(c_i, c_j)$ respectively,

$$t, r, a \in [1; 0], t + r + a = 1, S(c_i, c_j) \in [1; 0]$$

$$\begin{cases} S(c_i, c_j) = 1, \text{if concepts are equivalent}, \\ S(c_i, c_j) = 0, \text{if concepts are different.} \end{cases}$$

2.4 Genetic algorithm for finding weighting factors

To solve the problem of finding the weight coefficients, it is proposed to use a genetic algorithm that most effectively provides a solution for functions with several extrema. A modified genetic algorithm was used as the general structure of the algorithm.

The genetic algorithm is a heuristic algorithm that includes the principles of natural evolution and the idea of "survival of the fittest". The solution is encoded as a sequence of genes, which is called an individual.

The task of assessing the semantic proximity of ontology concepts belongs to the group of constrained optimization problems. Let's represent it as follows:

$$minf_{t,r,a}(\bar{x})\bar{x} = (t,r,a) \in F \subseteq S$$

$$t,r,a \in [0;1]$$

$$t+r+a=1,$$

$\bar{x}$ – a solution vector satisfying all constraints is called a feasible solution;

F – range of feasible solutions;

S – whole search area.

We formulate the optimization problem as follows: it is necessary to find $\bar{x'} \in F$, such that $f_{t,r,a}(\bar{x}) \geq f_{t,r,a}(\bar{x}) \forall \bar{x} \in F$

We construct a chromosome, which consists of a set of genes t, r, a. The objective function is based on the use of Euclidean distance:

$$f_{t,r,a} = \sum_{c_i \in O_1, c_j \in O_2} (t \cdot S^T(c_i,c_j) + r \cdot S^R(c_i,c_j) + a \cdot S^A(c_i,c_j) - 1)^2 \qquad (15)$$

1. Initialization

The initial population is given by a randomly generated set of values with the following restrictions $t,r,a \in [0;1\,, t+r+a=1$. The type of such a population is presented in Table 1, where n is the population size.

Table 1: General structure of the population

C1			C2			…	Cn		
t_1	r_1	a_1	t_2	r_2	a_2		t_n	r_n	a_n

The fitness function is the target function (15).

2. Selection

2.1 The coefficient of fitness is calculated for each chromosome.

2.2 Those chromosomes are selected that will participate in the creation of descendants by the method of tournament selection (Figure 3). All chromosomes are paired with the subsequent selection of the chromosome with the best fit. A deterministic choice is made with a probability of 2%. Subgroups are 2 individuals in size.

A new population is being formed, which consists of chromosomes obtained as a result of the application of genetic operators to the chromosomes of the parent population. The new population becomes current for this iteration of the genetic algorithm.

3. Crossover

The crossover is applied to a pair of chromosomes from the parent population. As a result of their recombination, a new generation is obtained. Further, for each pair of parents, a random gene position is determined as a crossing point in the chromosom

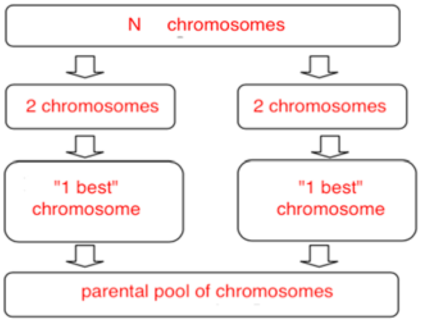

Figure 3: Scheme illustrating the tournament selection method for subgroups

A study was carried out on the example of ontology integration: information systems for managing the educational process and information systems for financial planning, as well as information systems "Progress" and information systems for managing the educational process, information systems "Auditorium" and information systems "Schedule". As a result of computational experiments, the most efficient genetic operators and parameters were determined. The analysis of the results obtained showed that GA gives the best result when using several crossover operators: 30% single-point crossover, 40% arithmetic crossover and 30% two-point crossover.

4. Mutation

In each new chromosome, a gene is randomly selected and mutated. The probability of a gene mutation is very small (~ 1-2%). All duplicate chromosomes are removed from the population.

The two-point mutation operator changes the value of two genes from t, r or a in the chromosome to random numbers in the range [0,1] subject to the following constraint $t+r+a=1$.

5. Assessment of the fitness of chromosomes in a population

At each iteration, the values of the fitness function are calculated for all chromosomes of the current population.

6. Checking the condition for stopping the algorithm

The algorithm stops if the following condition is met: $|f_{t,r,a_i} - f_{t,r,a_{i-1}}| < 1.0E - 6$

Another condition for stopping the genetic algorithm is a given execution time or a certain number of iterations.

7. Choosing the "best" solution

The result is recorded as a chromosome. It is chosen with the smallest value of the fitness function, otherwise we proceed to the next step - selection.

The block diagram of the algorithm is shown in Figure 4. In order to improve the efficiency of the use of genetic operators, the standard genetic algorithm has been modified. A number of genetic operators were included in the GA: selection, 30% one-point crossing over, 40% arithmetic crossing over, and 30% two-point; random mutation.

Below is an example of how crossover operators work.

Single point crossover:

X.-father: t_1 | r_1, a_1 X.-mother: t_2 | r_2, a_2 X.-descendant: t_1 | r_2, a_2 or t_2 | r_1, a_1

Point-to-point crossover:

X.- father: t_1 | r_1| a_1 X.- mother: t_2 | r_2| a_2 X.-descendant: t_2 | r_1|a_2 or t_1 | r_2| a_1

Arithmetic point crossover:

X.- father: t_1, r_1, a_1 X.- mother: t_2, r_2, a_2 X.-descendant:

w×t_1 +(1-w)×t_2, w×r_1 +(1-w)×t_2, w×a_1 +(1-w)×a_2

or

w×t_2 +(1-w)×t_1, w×r_2 +(1-w)*t_1, w×a_2 +(1-w)×a_1 ,

where w – constant, random number from the interval [0,1].

The use of a number of genetic operators identified in the experiment makes it possible to obtain a generation of individuals with the best value of the objective function and leads to an overall reduction in the time for solving the problem.

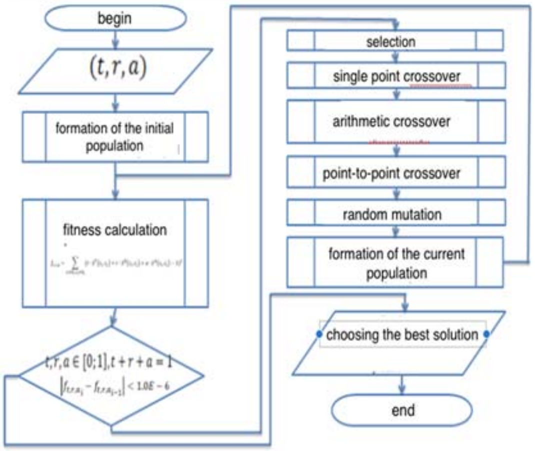
begin
(t,r,a)
formation of the initial population
fitness calculation
$t,r,a \in [0;1], t+r+a=1$
$|f_{t,r,a_i} - f_{t,r,a_{i-1}}| < 1.0E-6$
selection
single point crossover
arithmetic crossover
point-to-point crossover
random mutation
formation of the current population
choosing the best solution
end

Figure 4: Block diagram of the algorithm

The assessment of the reliability of the results of the genetic algorithm was carried out for the case of finding the concepts "Partially equivalent" when integrating financial planning information systems and HR management information systems. To do this, let's analyze the absolute and relative error ($Eabs$, $Erel$).

$$Eabs\ K, K_э = |K \cup K_э| - |K \cap K_э|$$

$$Erel\ K, K_э = Eabs\ K, K_э / |K_э|,$$

where $K_э$ - set of initial partially equivalent concepts obtained by an expert, $K_э = 57$;

K - set of partially equivalent concepts found by the algorithm $K \subseteq K_э, K = 42$.

As a result, the absolute error is 15, and the relative error is 0.26.

The degree of coverage cd by a set of partially equivalent concepts of the set of initial ones. For the set of found concepts, it is equal to $cd = 1 - Erel = 0.74$. It follows from this that the reliability of the found partially equivalent concepts is quite high.

A comparative analysis with the brute force method and the gradient descent method is carried out. When using the enumeration method with an increase in the number of concepts in the ontology, the number of solution options increases. The gradient descent method is based on the determination of the local maximum by choosing some random values of the parameters. By changing the values of these parameters, you can achieve the highest growth rate of the objective function. If the maximum is reached, then such an algorithm stops. Finding the global optimum will require additional efforts. This method does not guarantee the optimality of the found solution. As a result of the analysis, it was revealed that the proposed genetic algorithm has accelerated convergence and shows the best end result.

2.5 Classification of levels of proximity of concepts

The method for calculating the semantic proximity of concepts allows you to quantify the similarity between concepts. For each concept of one ontology, a set of relevant semantic concepts of another ontology is formed. In order to rank the elements of the result set, it is necessary to determine the threshold values of the proximity measure.

A method has been developed for classifying the levels of proximity of concepts to establish their correct display (Figure 5).

The question of finding the minimum threshold b of semantic proximity at which the concepts are assumed to be equivalent is considered.

$$b = max(S(c_i, c_j)|\forall c_i \in O_1, \forall c_i \in O_2 * p_1/100 \quad (16)$$

where p_1 – percentage at which b is taken as a similarity threshold for establishing equivalence and correct display c_i and c_j.

It is shown that b is the minimum threshold at which a decrease in this value leads to the impossibility of a complete display of ontology elements.

A threshold value is found at which the concepts are assumed to be partially equivalent.

$$q = maxS\ c_i, c_j \forall c_i \in O_1, \forall c_i \in O_2 * p_2/100 \ (17)$$

where p_2 –the percentage at which q is accepted as the similarity threshold for establishing partial equivalence of concepts.

It is shown that q is the minimum value in the sense that a decrease in this value leads to an incorrect display of ontology elements.

Concepts are accepted different if semantic closeness measures not exceeding the threshold q are important.

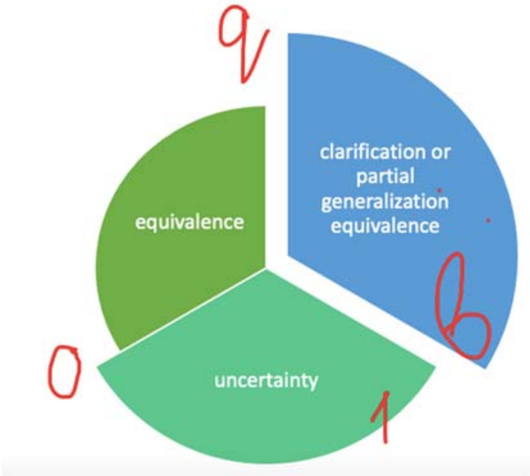

Figure 5: Method of classification of levels of semantic similarity of concepts

Thus, it is possible to construct a model of a single integrated information space based on the ontologies of information systems of different subject areas, which is a unified entry point for information from systems and data sources into a single information space.

The constructed model of a single integrated information space reflects information systems in the best way. The constructed model serves as a basis for determining semantic dependencies, and also makes it possible to apply the technology of integrating data from information systems of different subject areas.

The result of mathematical modeling is the construction of a model for the integration of information systems, as well as proof of its compliance with the set research goal. The applicability of the model was investigated when integrating systems of different subject areas of the university. According to the analysis of the obtained results, the constructed model of information systems integration is capable of adequately describing the initial situation. The integration algorithm using ontologies as a whole is free from many disadvantages inherent in purely technical methods, and provides an opportunity to develop integrated information systems that work with information at the semantic level.

2.6 Algorithm for the integration of information systems based on ontologies

The constructed mathematical model for displaying and integrating information systems ontologies well describes their semantic features.

The integration of the structures of the concepts of the initial ontologies is represented as the integration of the generic hierarchy, starting from the root of the hierarchical tree. This task, in turn, is convenient to perform recursively: integrate the root of the hierarchy (the incoming concept) into the base hierarchy (also specified by its root - the input parameter) and the subtrees of each of the subordinate concepts of this concept by calling the same procedure with the current subordinate concept of this concept as the top of the hierarchy. For the possibility of recursive treatment, this part of the system integration algorithm is separated into a separate procedure.

Integration of the concepts of the initial ontologies consists in placing the given concept of the integrated ontology into the hierarchy of the basic ontology. The general scheme of the algorithm operation is to insert an integrable class of concepts into the base hierarchy at the lowest possible level. Walking down the hierarchy is implemented recursively.

Using the constructed model and the method for assessing the semantic proximity of ontology concepts, as a result of a computational experiment, an information system integration algorithm was developed, which can be divided into six stages (Figure 6):

1. Comparison of ontologies.

A selection of the initial ontologies O and O' of integrable information systems is performed. It is assumed that O is the basic, basic ontology and O' is the integrable ontology. Weights are calculated for measures of semantic proximity of concepts, as well as threshold values for classifying relationships between concepts C.

2. Integration of concepts.

Step 1. The set C_1' of subordinate concepts with the root vertex C_1 of the basic ontology O and the set C_2' of subordinate concepts from the root C_2 of the basic ontology O and the set C_2' of the integrable ontology O are formed.

Step 2. Beginning of the cycle. The cycle compares and integrates the elements of the set C_2' with the elements of the set C_1', i.e. a concept from the set C_2' is integrated with the concept-vertex C_1 in the ontology hierarchy.

Step 3. The measure of semantic similarity is calculated for concepts from the set C_1' and concepts from the set C_2'.

Step 4. In accordance with the threshold values of the proximity measure, the type of semantic dependence between the concepts is determined and either a mapping is established or a conflict resolution algorithm is performed. The loop is executed until all elements of the set C_2 have been analyzed.

If dependencies z^1, z^5 are found, then the concept mapping is set automatically. If the dependences z^2, z^3, z^4 are found, then the correctness of the constructed mapping is confirmed manually.

3. Checking the result. Checking the correctness of the resulting ontology.

4. Interpretation. Derivation of the resulting mappings between concepts and attributes of information systems ontologies.

5. Iteration. Repetition of some steps of the algorithm.

6. Establishment of mappings between the elements of object schemes of information systems based on the connection of ontological concepts. After that, it becomes possible to generate requests for adding information from one information system to another.

At the first stage of the algorithm, variants of the mutual relationship of concepts are taken into account. The result of the operation of their comparison leads to five different operations on concepts.

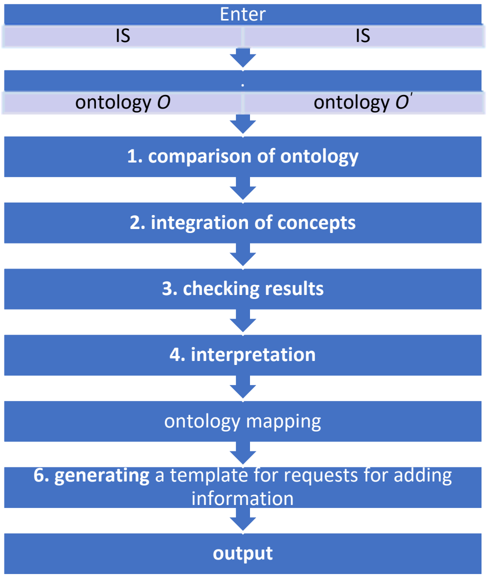
Enter
IS
IS
.
ontology O
ontology O'
1. comparison of ontology
2. integration of concepts
3. checking results
4. interpretation
ontology mapping
6. generating a template for requests for adding information
output

Figure 6: The process of integrating information systems

1. If the concepts are equivalent, then they represent the same concept in the ontology, therefore, must be "glued" into one.

2. If the concept of one ontology is a generalization of the corresponding concept of another ontology, such concepts should be represented as a concept and a subclass, respectively, and the matching attributes should be removed from the subclass, since they will be inherited from the

superclass (since the class-subclass relationship is a partial ordering).

3. If the concept of one ontology is a refinement of the corresponding concept of another ontology, such concepts should be represented as a subclass and a class, respectively, and the matching attributes should be removed from the subclass, since they will be inherited from the superclass. Here it is necessary to take into account all the existing relationships of these concepts.

4. If the concepts of two ontologies are partially equivalent, then they represent similar concepts, that is, they must have a common superclass, which is their generalization (note that this superclass was not present in any of the original ontologies), while the matching attributes must be removed from the subclass since they will be inherited from the generic concept.

5. One ontology may lack an equivalent concept from another ontology. The situation when the measure of proximity of concepts is less than the threshold at which it can be considered that the concept is absent in the original ontology. In this case, measures of semantic closeness between the given concept and all the concepts of the original ontology are calculated. Concepts are selected for which the measure of proximity S is maximal, and the mapping into the resulting ontology is set. If the condition is met, then the concept is copied into the resulting ontology with attributes and relationships as a subclass of the concept that has a mapping.

After building the resulting ontology of information systems, it becomes possible to interpret information from one information system by means of another information system.

The result of mathematical modeling in this article is the construction of a model for the integration of information systems.

The constructed models of integration of information systems are able to adequately describe the initial situation. The integration algorithm using ontologies as a whole is free from many of the drawbacks inherent in purely technical methods, and provides an opportunity for the development of integrated information systems that work with information at the semantic level.

3. CONCLUSION

The study received a mathematical model of integration of information systems with heterogeneous ontological specifications that analyzes the semantic connections arising between similar elements ontology information systems integrable.

As well as a computational method for determining the semantic proximity of concepts has been developed, as well as a method for classifying the levels of their proximity in order to build a resulting (integrated) ontology.

Also, a genetic algorithm is proposed that has accelerated convergence and shows the best end result.

Scope of the results obtained:

- to implement an on-demand data integration approach when there are no special hardware requirements;
- for cases when the problem of access to fresh data becomes more urgent than the systematization of already accumulated data;
- in the process of the IAIS development, if there is a need to change the schemes, data models of the integrated subsystems.

The results obtained can be applied to implement an approach to data integration on demand.

REFERENCES:

[1] Ehrig M., Sure Y. // The semantic web: Research and applications. Proc. 1st European Semantic Web Symposium. LNCS. Berlin: Springer, 2004. V. 3053. p. 76.

[2] Tuzovskiy, A.F. Ontologo-semanticheskiye modeli v korporativnykh sistemakh upravleniya znaniyami: dissertatsiya doktora tekhnicheskikh nauk. Tomskiy politekhnicheskiy universitet, Tomsk, 2007.

[3] Shakhgel'dyan, K.I. Teoreticheskiye printsipy i metody povysheniya effektivnosti obrazovatel'nykh uchrezhdeniy na osnove ontologicheskogo podkhoda: dissertatsiya doktora tekhnicheskikh nauk. In-t

[4] Wu Z., Palmer M. // Proc. 32nd Annual Meeting of the Association for Comput. Linguistics. Las Cruces, 1994. P. 133.

[5] Nguyen H.A. Thesis for the Degree Master of Science. – University of Houston−Clear Lake, 2006.

[6] Resnik P. Using information content to evaluate semantic similarity in a taxonomy // Proc. 14th Int. Joint Conf. on Artificial Intelligence. Montreal, 1995. P. 448.

[7] Henrik Bulskov, Rasmus Knappe, Troels Andreasen. //. Proc. 5th Int. FQAS Conf. LNCS. V. 2522. P. 100. Berlin: Springer, 2002.

[8] Maedche A., Staab S. // Proc. 13th EKAW Conf. LNAI. Berlin: Springer, 2002, − P. 251.

[9] Rodríguez M.A. Thesis for Degree of Doctor of Philosophy. University of Maine, 2000.

[10] Falkl J., High R., Lau Ch. Service Oriented Architecture Compliance: Initial steps in a longer journey.− 2005.

[11] Hao He. What is Service-oriented Architecture. −2003.−http://www.xml.com/pub/a/ws/2003/09/30/soa.html (data obrashcheniya: 01.12.2020).

[12] Stojanovic N., Madche A., Staab S. et al. // Proc. 1st Int. Conf. on Knowledge Capture. New York, 2001. P. 155.

[13] Guarino N. Formal ontology, conceptual analysis and knowledge presentation//International Journal of Human and Computer Studies, 43(5/6), pp. 625−640.

[14] Guarino N. Formal Ontology in Information Systems, Proceedings of FOIS'98, Trento, Italy, 6−8 June 1998. Amsterdam, IOS Press, pp. 3−15.

[15] Gruber T.R. A Translation Approach to portable ontology specification//Knowledge Systems 92−7, Laboratory, Stanford University, Technical Report KSL.−1993.

[16] Kalinichenko L.A. Methods and tools for equivale nt data model mapping construction. Advances in Database Technology: Proc. of the International Conference on Extending Database Tec hnology EDBT'90. LNCS 416. − Berlin-Heidelberg: Springer-Verlag, 1990. − P. 92−119.

[17] Vernikov G. Standarty ontologicheskogo issledovaniya IDEF5; URL: www.vpg.ru/main.mhtml?PubID=25 (data obrashcheniya: 18.12.2020).

[18] Hirst G., St-Onge D. // WordNet: An electronic lexical database. Cambrige, 1998. P. 305.

[19] Leacock C., Chodorow M. // WordNet: An electronic lexical database. Cambrige, 1998. − P. 265.

[20] T.Temirbolatova. R.Uskenbayeva, Young Im Cho, Z.Uskenbayeva, G.Bektemyssova, A.Kassymova. Recursive decomposition as a method for integrating heterogeneous data sources//Proceedings of the 15th International Conference on Control, Automation and Systems (ICCAS 2015). – Busan, South Korea. October 13-16, 2015 – P.2076-2079. ISSN: 2093 – 7121

[21] Wu Z., Palmer M. // Proc. 32nd Annual Meeting of the Association for Comput. Linguistics. Las Cruces, 1994. P. 133.

[22] Nguyen H.A. Thesis for the Degree Master of Science. – University of Houston−Clear Lake, 2006.

[23] Resnik P. Using information content to evaluate semantic similarity in a taxonomy // Proc. 14th Int. Joint Conf. on Artificial Intelligence. Montreal, 1995. P. 448.

[24] R. Uskenbayeva, T. Temirbolatova, Y. Chinibayev, A. Kassymova, K. Mukhanov. Technology of integration of diverse databases on the example of medical records//Proceedings of the 14th International Conference on Control, Automation and Systems (ICCAS 2014) - Gyeonggi -do, Korea, 2014. P 282-285. ISSN: 2093- 7121.

[25] R.Uskenbayeva, T.Chinibayeva. Algorithm for the construction of an ontology in the field of scientific knowledge//The Bulletin of Kazakh Academy of Transport and Communications named after M. Tynyshpayev ISSN 1609-1817. Vol. 107, No.4 (2018), pp. 259-266

[26] R.Uskenbayeva, T.Chinibayeva. Method of extracting meta description from databases//Herald of the Kazakh-british technical university ISSN1998-6688. Vol.15, No.4 (2018), pp. 116-123

[27] T. Chinibayeva. Security semantic database problems//Herald of the Kazakh-British technical university ISSN1998-6688. Vol.16, No.3 (2019), pp. 168-174

[28] Uskenbayeva, A. Kuandykov, Young Im Cho, T. Temirbolatova, S.Amanzholova, D.Kozhamzharova Integrating of data using the Hadoop and R. The Procedia Computer Science, ISSN: 1877-0509, Vol: 56, Issue: 1, Page: 145-149