Building a Comprehensive Digital Archive for the Polotsk Cadet Corps: A Case Study Intelligent Information and Reference System Design

Arkadzi Oskin
Euphrosyne Polotskaya State University of Polotsk
Polotsk, Belarus
Email: a.oskin@psu.by

***Abstract*—On the example of designing and building an Intelligent Information Retrieval System (IIRS) "Polotsk Cadet Corps", the main stages of creating an IIRS are described. The principles of ontological modeling and the tools used for these purposes are briefly considered. Based on the analysis of the subject area, an ontological model is built, which can be transformed into a working IIRS by using the OSTIS technology stack. The advantages of OSTIS technologies for solving problems of this kind are described.**

***Keywords*—OSTIS Technology, ontological modeling, reference system, intelligent system**

I. Introduction

The Polotsk Cadet Corps was opened on July 8 (June 25, old style) 1835 by order of Emperor Nicholas I. The building was located in the building of the former Jesuit Academy. Currently, this historic building houses the Faculty of Information Technology and the Faculty of Humanities of Euphrosyne Polotskaya State University of Polotsk, which explains our interest in the history of the Polotsk Cadet Corps.

In order to create an Intelligent Information and Reference System, we built an ontological model of the subject area. Ontological modeling is a method of modeling information systems that focuses on the relationships between concepts and entities. It is a branch of artificial intelligence that is concerned with representing knowledge in a way that is intuitive and easy to understand. Ontological modeling is used in a variety of applications, including information retrieval, natural language processing, and semantic web technologies. In the context of intellectual information systems, ontological modeling is used to represent knowledge in a way that can be easily understood by humans and machines alike.

There are several steps involved in ontological modeling in Intellectual Information Systems:

1) Define the domain: Identify the domain of the system, including the concepts and entities that are relevant to it. This involves understanding the context of the system and its intended use.
2) Create a conceptual model: Develop a conceptual model of the domain, which includes the concepts and relationships between them. This can be done using a variety of modeling techniques, such as UML, ER diagrams, or other graphical notations.
3) Define the ontology: Create an ontology that formalizes the conceptual model, using a language such as OWL or RDF. This involves specifying the relationships between concepts in a way that can be understood by both humans and machines.
4) Populate the ontology: Populate the ontology with data, using tools such as Protégé or other ontology editors. This involves creating instances of concepts and linking them together in a way that reflects the relationships defined in the ontology.
5) Use the ontology: Once the ontology has been populated, it can be used to support a variety of applications, such as information retrieval, natural language processing, or semantic web technologies.

Ontological modeling can be a complex process, but it is a powerful tool for representing knowledge in information systems. By using ontological modeling, it is possible to create systems that are more intuitive and easier to understand, both for humans and machines.

There are several advantages of using ontological modeling in intellectual information systems, including:

1) Automated reasoning: By having the essential relationships between concepts built into them, ontologies enable automated reasoning about data. This makes it easy to implement semantic graph databases that use ontologies as their semantic schemata.
2) Increased use, reuse, and maintainability: Ontologies facilitate domain knowledge sharing, with a common vocabulary across independent software

applications. This leads to increased use, reuse, and maintainability of the information systems.

3) Easy to relate to object-oriented programming paradigm or database design: Those who are familiar with the object-oriented programming paradigm or database design can easily relate the ontological representation of the domain entities to classes or database schemas. The classes are generic representations of the entities that encapsulate properties and behaviors.
4) Semantic model of data: One major advantage of using a domain ontology is its ability to define a semantic model of the data combined with the associated domain knowledge. Ontologies can also be used to define links between different types of semantic knowledge. Thus, ontologies can be used in formulating some data searching strategies.

Overall, ontological modeling in intellectual information systems provides a powerful tool for representing knowledge in a way that is intuitive and easy to understand, both for humans and machines.

Ontology editors are applications designed to assist in the creation or manipulation of ontologies. They use one or more ontology languages to create, visualize, and manipulate ontologies. These editors have features such as visual navigation possibilities within the knowledge model, inference engines and information extraction, support for modules, the import and export of foreign knowledge representation languages for ontology matching, and support of meta-ontologies such as OWL-S and Dublin Core. Additionally, there are various tools used for ontological modeling of intelligent information systems. Let's consider the most popular of ontology editors.

II. Ontology Editors

To create ontologies, specialized software products are widely used - ontology editors. Let's look at the most popular of them. We will compare the characteristics of the following software products.

There are several popular ontology editors that can help authors create their ontologies. Some of them provide additional functions and plugins that can be useful when working with ontologies. Some of the most popular ontology editors are listed below:

1) NeOn Toolkit is an ontology editor with many plug-ins available, especially suitable for large scale projects (eg multi-module ontologies, multilingual, ontology integration, etc.).
2) Neologism is an online dictionary editor and publishing platform.
3) Vitro is an integrated editor for ontologies and semantic web applications.
4) Knoodl is a community-oriented ontology and knowledge base editor.
5) Fluent Editor is a comprehensive tool for editing and manipulating complex ontologies that uses controlled natural language. It provides an instant natural language representation of OWL/SWRL, which improves performance and makes editable ontologies easier to read and understand.

Some other popular ontology editors that can be mentioned include:

1) Eddy
2) OntoME
3) OntoStudio (formerly known as OntoEdit)
4) Protégé

Let's take a closer look at some of the editors from the list above.

NeOn Toolkit.

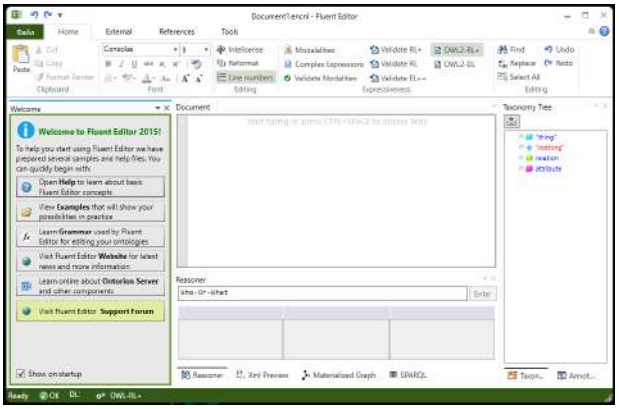
NeOn Toolkit is an ontology editor that allows users to create and edit ontologies. It offers a variety of tools and features to support the development of ontologies, including visual modeling, ontology debugging, and ontology testing. Some of the advantages of using NeOn Toolkit are:

- User-friendly interface: NeOn Toolkit has an intuitive interface that makes it easy to create and edit ontologies.
- Visual modeling: The tool offers a visual modeling environment that allows users to create and edit ontologies using graphical representations.
- Collaboration: NeOn Toolkit supports collaboration between users, making it easy to work on ontologies as a team.
- Ontology debugging: The tool provides debugging capabilities that help users identify and fix errors in their ontologies.
- Ontology testing: NeOn Toolkit includes a testing framework that allows users to test their ontologies to ensure they are working correctly.
- Ontology testing: NeOn Toolkit includes a testing framework that allows users to test their ontologies to ensure they are working correctly.

Despite its advantages, NeOn Toolkit also has some drawbacks. For example:

- Steep learning curve: While NeOn Toolkit is user-friendly, it can still be challenging to learn for users who are new to ontology development.
- Limited documentation: Some users have reported that the documentation for NeOn Toolkit is limited, making it difficult to troubleshoot issues or learn about advanced features.
- Limited support: NeOn Toolkit is an open-source tool, which means that support is limited to user forums and community resources.

NeOn Toolkit is primarily used in the field of semantic web development, specifically in the development of ontologies. It is commonly used in research and academic



Welcome to Fluent Editor 2015!

To help you start using Fluent Editor we have prepared several samples and help files. You can quickly begin with:

Open **Help** to learn about basic Fluent Editor concepts

View **Examples** that will show your possibilities in practice

Learn **Grammar** used by Fluent Editor for editing your ontologies

Visit Fluent Editor **Website** for latest news and more information

Learn online about **Ontorion Server** and other components

Visit Fluent Editor **Support Forum**

☑ Show on startup

Start typing or press CTRL+SPACE to display hints

Who-Or-What

Figure 1. Fluent Editor.First start.

settings for ontology development, but it can also be used in industry settings for knowledge management and data integration.

Fluent Editor.

Fluent Editor is an ontology editor developed by the Polish company Cognitum. It is used for editing complex ontologies created using a controlled natural language (CNL). The editor allows users to create ontologies by entering phrases in natural language.

Some of the advantages of Fluent Editor are:

- User-friendly interface: Fluent Editor has a simple and intuitive graphical user interface that allows even non-experts to create and edit ontologies.
- Support for natural language: Fluent Editor supports natural language input, which makes it easier to create ontologies for non-experts.
- Advanced features: Fluent Editor includes advanced features such as automated reasoning, which can help identify inconsistencies and errors in ontologies.

However, there are also some limitations to Fluent Editor:

- Limited documentation: Fluent Editor has limited documentation available, which can make it difficult for users to learn how to use the editor.
- Limited support: Cognitum provides limited support for Fluent Editor, which can make it difficult for users to get help if they encounter problems.
- Limited customization: Fluent Editor has limited support for customization, which can make it difficult for users to tailor the editor to their specific needs.

Fluent Editor is primarily used for creating and editing complex ontologies [0][2][4]. It can be used in various domains, including healthcare, finance, and engineering, where ontologies are used to represent and organize knowledge.

Eddy.

Eddy is an ontology editor that allows users to create and edit ontologies. Ontologies are used to define concepts and relationships in a specific domain of knowledge. Eddy is a web-based application that can be used by anyone with an internet connection. It has several advantages, including:

- User-friendly interface: Eddy has a simple and intuitive interface that makes it easy for users to create and edit ontologies.
- Collaboration: Eddy allows multiple users to work on the same ontology simultaneously, making it ideal for collaborative projects.
- Integration: Eddy can be integrated with other tools and applications, making it a versatile tool for ontology development.

However, Eddy also has some limitations:

- Limited functionality: Eddy's functionality is limited compared to other ontology editors.
- Learning curve: Although Eddy is user-friendly, there is still a learning curve for users who are new to ontology editing.
- Lack of support: Eddy does not have a large community of users, which means that there is limited support available for users who encounter problems.

Eddy is primarily used in the field of knowledge engineering, which involves the creation and management of knowledge-based systems. It can be used in a variety of domains, including:

- Healthcare: Eddy can be used to create ontologies for medical terminology and patient data.
- E-commerce: Eddy can be used to create ontologies for product catalogs and online marketplaces.
- Education: Eddy can be used to create ontologies for educational resources and curriculum development.

Overall, Eddy is a useful tool for ontology editing, but it may not be the best option for all users depending on their specific needs and requirements.

OntoME.

OntoME is an ontology editor that enables users to create, edit and publish ontologies. Here are some advantages and disadvantages of ontoME:

Advantages:

- OntoME provides a user-friendly interface for creating and editing ontologies, allowing users to create ontologies without programming knowledge.
- It supports multiple ontology formats, such as OWL, RDF, and RDFS.
- It offers a range of features, such as the ability to import and export ontologies, search for terms, and visualize ontologies.
- OntoME allows for collaboration and sharing of ontologies through a web-based interface.
- It provides support for versioning and change management of ontologies.

Disadvantages:

- OntoME may not be suitable for heavy-weight projects, such as multi-modular ontologies or ontology integration, as it does not offer advanced features for such projects.
- It may not be suitable for users with advanced programming knowledge, as it does not allow for direct editing of the ontology code.
- The web-based interface may have limitations in terms of performance and speed compared to desktop-based ontology editors.

The scope of OntoME is to provide a user-friendly and accessible ontology editor for users with varying levels of ontology development expertise. It is suitable for small to medium-sized ontology projects with moderate complexity. For more complex projects, users may need to consider other ontology editors that offer advanced features and customization options.

OntoStudio.

OntoStudio is an ontology editor that provides a professional environment for ontology development. It supports W3C standards such as OWL, RDF, and RDFS, and F-Logic for the logic-based processing of rules. OntoStudio also comes with many connectors to databases, documents, file systems, applications, and web services.

The modular design of OntoStudio enables users to enrich it with self-developed modules and customize it according to their personal needs. It has modeling tools for ontologies and rules, as well as components for the integration of heterogeneous data sources. OntoStudio is available with a free evaluation license.

In a study that compared five ontology editors, including OntoStudio, the main criterion for comparison was the convenience for users. The study described the basic features and structure of the editors, as well as their way of use. OntoStudio was found to be a convenient tool for users.

OntoStudio's advantages are its support for various ontology languages and its modular design that allows customization. Its disadvantages are not mentioned in the sources. OntoStudio's scope is in the development and operation of semantic applications that involve ontology learning, reasoning, and text mining. It is also useful for the storage and management of semantic data and metadata.

Specific features of OntoStudio's modeling tools for ontologies and rules are not explicitly mentioned in the sources. However, OntoStudio combines modeling tools for ontologies and rules and is built on top of a powerful internal ontology model that allows the user to edit a hierarchy of concepts or classes. OntoStudio also supports W3C standards such as OWL, RDF, and RDFS, and F-Logic for the logic-based processing of rules.

The user-friendliness of OntoStudio for beginners in ontology modeling is not directly mentioned in the sources. However, a study that compared five ontology editors, including OntoStudio, found that the main criterion for comparison was the convenience for users. The study described the basic features and structure of the editors, as well as their way of use. OntoStudio was found to be a convenient tool for users. It is worth noting that OntoEdit is considered simpler than OntoStudio but lacks some important features as apprenticeship on ontologies grows.

After reviewing the editors discussed above, we came to the conclusion that the Protégé editor would be the most suitable for our purposes. Consideration of the methodology for creating an ontology in this editor is devoted to the next section of our work.

III. Ontology Editor Protégé

Protégé is an open-source ontology editor that allows users to create, edit, and manipulate ontologies. An ontology is a formal representation of a domain's knowledge that specifies a set of concepts and their relationships. Creating an ontology involves several steps, including defining the domain, identifying the key terms, and creating the class hierarchy.

To create an ontology in Protégé, one can follow these steps:

1) Define the domain: The first step is to determine the scope of the ontology, including the types of questions it should answer and the purpose it serves.
2) Identify key terms: After defining the domain, the next step is to identify the key terms that will be used in the ontology. This can be done by analyzing literature or consulting with experts. These terms should be organized into a table, including their properties or characteristics.
3) Create the class hierarchy: The next step is to create the class hierarchy, which involves identifying the most general concepts and gradually refining them into more specific ones. There are several approaches to creating a class hierarchy, including top-down, bottom-up, and combined approaches.
4) Define properties and characteristics: Once the class hierarchy is established, the next step is to define the properties and characteristics of each class. These may include attributes such as weight, habitat, and population size.
5) Add instances: Finally, instances of each class can be added to the ontology. These instances are specific examples of the classes in the ontology.
6) In addition to creating an ontology, Protégé also allows for merging ontologies and performing operations on classes, such as defining equivalent or inverse classes and transitive properties.

The advantages of using Protégé to create an ontology are:

1) Open-source and free: Protégé is a free and open-source ontology editor, which makes it accessible to anyone who wants to use it.
2) User-friendly interface: Protégé has a user-friendly interface that allows users to easily create, edit, and manipulate ontologies.
3) Strong community support: Protégé has a strong community of academic, government, and corporate users who use it to build knowledge-based solutions in various areas such as bio-medicine, e-commerce, and organizational modeling. This support ensures that Protégé is continually updated and maintained.
4) Customizable: Protégé allows users to customize their ontologies by adding new classes and properties and modifying existing ones.
5) Integration with other tools: Protégé can be integrated with other tools such as reasoners and visualization plugins, which makes it a powerful tool for ontology development.
6) Reasoning capabilities: Protégé has built-in reasoning capabilities that allow users to check the consistency of their ontologies and detect errors.
7) Overall, Protégé is a powerful and flexible tool for creating and managing ontologies, with a user-friendly interface, strong community support, and a range of customizable features.

IV. Ontological Model

The domain ontological model was built using the Protégé editor. We have identified the following three classes: "Documents", "Events", "People". The "TextDocuments", "Pictures", "Video" and "Audio". Class "People" — subclasses "Graduates", "Directors", "Mentors". The "Events" class is subclasses corresponding to the reign of one or another director of the corps.

The structure of the upper level of the ontology "History of the Polotsk Cadet Corps" is shown in Figure 1.

V. Creation of the Intelligent Information and Reference System

We plan to complete the creation of the Intellectual Information and Reference System "History of the Polotsk Cadet Corps" using the OSTIS technology stack. We list the advantages of this approach, which guarantee the success of the completion of the project as a whole.

1) Any OSTIS — system can be easily supplemented with new knowledge or new methods for solving problems. This allows you to quickly and easily repurpose the developed intelligent system, reorienting it to a new range of tasks to be solved.
2) OSTIS — the system is focused on the reuse of the developed components. Thanks to a single and universal SC code, a library of typical components can be created, the use of which in the design process can reduce development time by 40-60 percent.
3) OSTIS — the system is reflexive, i.e. can analyze itself, due to the fact that it is fully described using the SC code. Reflexivity is one of the most important qualities of intelligent systems.
4) Due to the fact that the design of OSTIS — the system is reduced to the construction of its SC-model, it is completely platform-independent, and can be implemented both in software and in hardware.
5) Hardware implementation involves the creation of a new generation of computing devices — semantic computers.

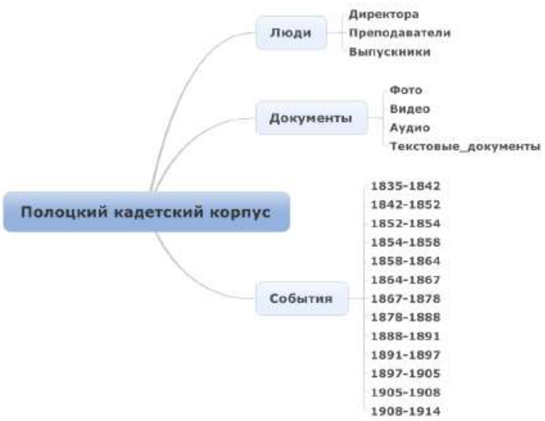


Figure 2. The structure of the upper level of the ontology "History of the Polotsk Cadet Corps"

VI. Conclusion

In conclusion, we note the following. Ontological modeling is an important method for developing intelligent information and reference systems. By focusing on relationships between concepts and entities, this approach ensures that knowledge is represented in an intuitive and machine-readable way.

With the help of ontological modeling, one can define a subject area, develop a conceptual model, create and populate an ontology with data, and finally put the ontology into action. Although it can be a complex process, the benefits of ontology modeling are clear: it provides a foundation for building powerful applications that address a variety of application problems.

An effective tool for building ontologies is the Protégé editor. It allows you to create classes, slots and instances, and also provides an easy way to modify ontologies without creating inconsistent data and knowledge. Protégé can be used for practical applications, such as creating ontologies for intelligent information retrieval systems, as well as for educational purposes, such as creating ontology models for e-learning.

Note also that Protégé can be easily integrated with other software used to work with ontologies. Based on the ontology created by Protégé tools, using the OSTIS technology stack, you can quickly create an Intelligent Information Retrieval System or a digital archive of an organization.

Returning to the IIRS "History of the Polotsk Cadet Corps", we note that the electronic resource created as a result of the above steps can be used in at least three organizations. In the Euphrosyne Polotskaya State University of Polotsk — as an exhibit of the Museum of the History of Science and Education of Polotsk, in the Polotsk Cadet Corps — as an exhibit of the corps museum being created, as well as in the exposition of the Polotsk National Historical and Cultural Museum-Reserve . In addition, the resource will certainly find application in the educational process of Polotsk State University.

Acknowledgment

The author expresses his deep gratitude to Professor V. V. Golenkov, thanks to whose advice and support I decided to write this article.

References

[1] V. V. Golenkov and N. A. Guljakina Cemanticheskaja tehnologija komponentnogo proektirovanija sistem, upravljaemyh znanijami. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, Minsk, BGUIR, 2015, pp. 57–78 (In Russ.)

[2] A. Maedche, B. Motik, L. Stojanovic, R. Studer, R. Volz Ontologies for enterprise knowledge management, *IEEE Intelligent Systems*, vol. 18, No. 2, IEEE, 2003, pp.26–33.

[3] R. S. Oktari, K. Munadi, R. Idroes, H. Sofyan Knowledge management practices in disaster management: Systematic review. *International Journal of Disaster Risk Reduction*, vol. 51, Elsevier, 2020, pp. 10–18.

[4] V. A. Carriero, A. Gangemi, M. Aldo, M. L. Mancinelli, L. Marinucci, A. G. Nuzzolese, V. Presutti, C. Veninata. ArCo: The Italian cultural heritage knowledge graph. The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18, Springer, 2019, pp. 36–52.

[5] M. L. Turco, M. Calvano, E. Giovannini Data modeling for museum collections. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, No. 2, 2019.

[6] A. Vlachos, M. Perifanou, A. Economides A review of ontologies for augmented reality cultural heritage applications. *Journal of Cultural Heritage Management and Sustainable Development*, Emerald Publishing Limited, 2019.

[7] M. N. Buharov Informacionnaja sistema dlja ontologicheskogo modelirovanija predmetnyh oblastej [Information system for ontological modeling of subject areas]. *Journal of Cultural Heritage Management and Sustainable Development*, Informatika, vol. 19, No. 2, 2022 (In Russ.)

[8] S. P. Vorona, E. O. Savkova Ispolzovanie ontologicheskogo modelirovanija pri razrabotke intellektual'noj sistemy dostupa k uchebno-metodicheskoj informacii [The use of ontological modeling in the development of an intelligent system for access to educational and methodological information]. Informatika, upravljajushhie sistemy, matematicheskoe i komp'juternoe modelirovanie (IUSMKM-2020), 2020, pp. 208–213(In Russ.)

[9] E. Kalashnikov, V. Pavlenko Primenenie ontologicheskogo modelirovanija v raznyh otrasljah [Application of ontological modeling in different industries], XXV Nauchno-tehnicheska konferencija s mezhdunarodnym uchastiem sbornik dokladov, P. 97 (In Russ.)

[10] J. A. Balyberdin Ontologicheskoe modelirovanie predmetnyh znanij v vuzah, konstruktorskih bjuro i nauchno-issledovatel'skih institutah [Ontological modeling of subject knowledge in universities, design bureaus and research institutes], 2022 (In Russ.)

[11] B. Reitemeyer, H.-G. Fill Enterprise, Business-Process and Information Systems Modeling: 20th International Conference, BPMDS 2019, 24th International Conference, EMMSAD 2019, Held at CAiSE 2019, Rome, Italy, June 3–4, 2019, Proceedings 20, Springer, 2019, pp. 212–226

[12] Z. A. A. Salam, R. A. Kadir, A. Azman Ontology Merging Using PROTÉGÉ–a Case Study. *Journal of Information System and Technology Management*, JISTM, 2021

[13] C. Shimizu, K. Hammar, P. Hitzler Modular graphical ontology engineering evaluated. The Semantic Web: 17th International Conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, Proceedings 17, Springer, 2019, pp. 20–35

[14] E. V. Glazyrin, T. E. Sohor Polockij kadetskij korpus v otkrytyh pis'mah, vypushhennyh «Obshhestvom vzaimopomoshhi polochan» v nachale XX veka [Polotsk cadet corps in open letters issued by the Polokhan Mutual Aid Society at the beginning of the 20th century], 2019 (In Russ.)

[15] S. G. Ljutko Stanovlenie i razvitie voennogo obrazovanija na zemljah Belarusi v XVIII–nachale XX veka [Formation and development of military education on the lands of Belarus in the XVIII — the beginning of the XX century], Minskij gosudarstvennyj lingvisticheskij universitet, 2021 (In Russ.)

[16] S. Poljakov and others Polockij kadetskij korpus. Istorija v licah [Polotsk cadet corps. History in faces], Matjeryjaly navukova-praktychnaj kanferjencyi: pa vynikah navukova-dasledchaj raboty 2008 g, NPIKZ, 2009, pp.48–59 (In Russ.)

[17] V. P. Vikent'ev Polockij kadetskij korpus. Istoricheskij ocherk 75-letija ego sushhestvovanija [Polotsk cadet corps. Historical sketch of the 75th anniversary of its existence], Polock: Tip. XV Kljachko, 1910 (In Russ.)

[18] V. V. Golenkov, N. A. Gulyakina, I. T. Davydenko, D. V. Shunkevich Semantic technologies of intelligent systems design and semantic associative computers. *Doklady BGUIR*, 2019, No. 3, pp. 42–50

[19] V. V. Golenkov, N. A. Gulyakina, V. A. Golovko, V. V. Krasnoproshin Methodological problems of the current state of works in the field of Artificial intelligence. *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, 2021, vol. 5, pp. 17–32

Создание комплексного цифрового архива полоцкого кадетского корпуса: пример проектирования интеллектуальной информационно-справочной системы

Оськин А. Ф.

На примере проектирования и построения Интеллектуальной информационно-поисковой системы (ИИПС) «Полоцкий кадетский корпус» описаны основные этапы создания ИИПС. Кратко рассмотрены принципы онтологического моделирования и инструменты, используемые для этих целей. На основе анализа предметной области строится онтологическая модель, которая может быть преобразована в работающую ИИПС с использованием стека технологий OSTIS. Описаны преимущества технологий OSTIS для решения задач подобного рода.

Received 27.03.2023