

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 plugins 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 userfriendly, 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

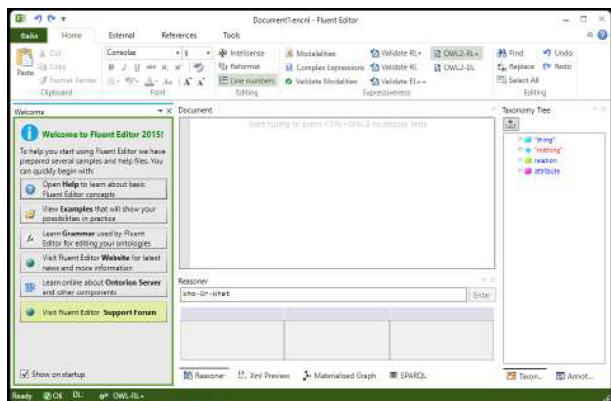


Figure 1. Fluent Editor.First start.

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: