ArchiGraph.Mir
Ontology Editor

User Manual

Version 1.12

https://archigraph.pro/

https://trinidata.com/

Contents

Preface

Welcome to ArchiGraph.Mir ontology editor!

ArchiGraph.Mir ontology editor is designed to create ontological models that can be used for a wide range of applications. To work with it, it is important to understand the concepts of ontological modeling.

The ontology created by ArchiGraph.Mir is saved in RDF triple store managed by ArchiGraph Platform and accessible via API ArchiGraph.MDM.

Individual objects (Individuals) of the ontology can be stored in any repositories managed by ArchiGraph.MDM including relational or document-oriented databases. The editor accesses the data using the ArchiGraph.MDM API, and therefore from the user's point of view it does not matter in which storage the data is located.

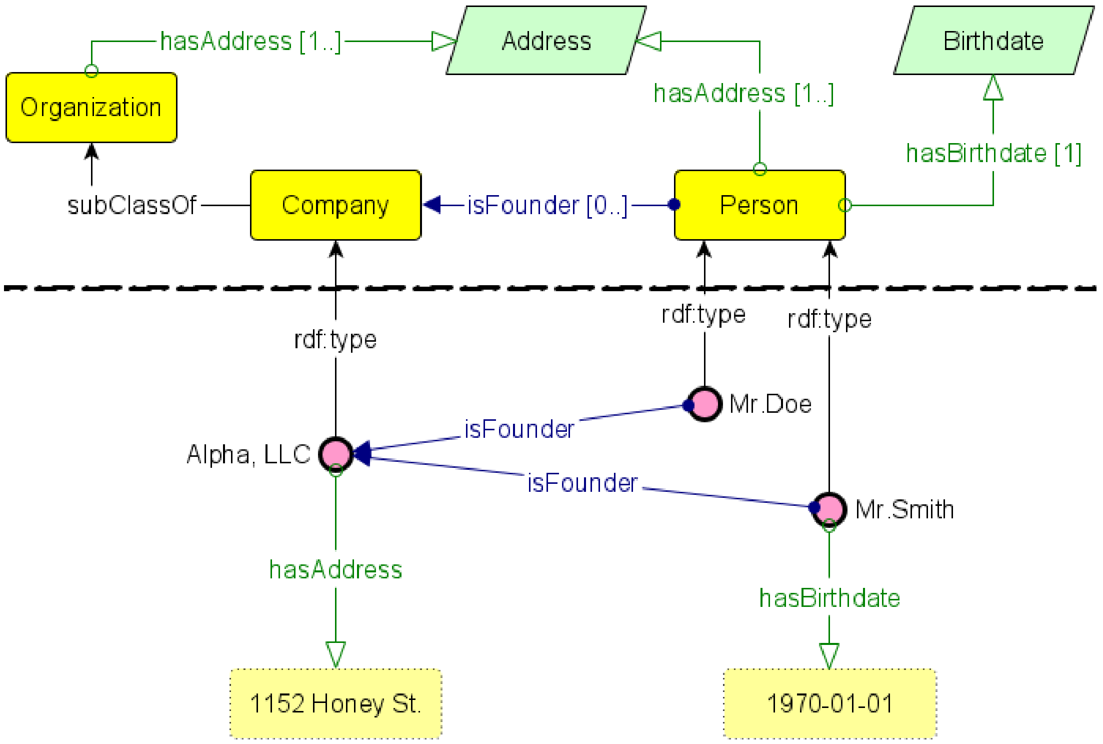
The ontology built in ArchiGraph.Mir can be exported to an RDF/XML or Turtle file or be used directly via the SPARQL endpoint provided by ArchiGraph.MDM.

Related Documents

The documentation for this product is available at: https://archigraph.pro/

Sample Ontology

All examples in this User Manual are discussed on the following ontology:



At the top of the diagram (TBox, Terminology Box) three classes are shown: "Person", "Organization", and "Company" as a subclass of organization. Organizations and persons have the `hasAddress` datatype property (or literal), which can take any number of values. The person also has a `hasBirthdate` property that has only one value for each person. Each company can have several person-founders. These relations are expressed by the property `isFounder`, the type of values for which are objects of the class “Company”.

At the bottom of the diagram (ABox, Assertions Box) the instances and property values are shown. Alhpa, LLC is located at 1152 Honey St. Mr.Doe and Mr.Smith are its founders. The birthdate of Mr.Smith defined as 1970-01-01.

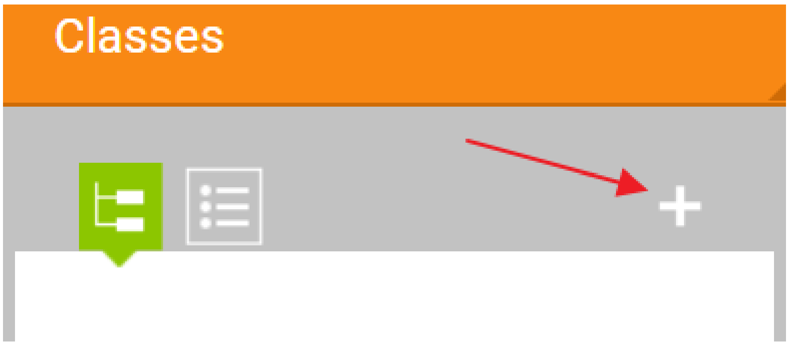
Core Operations with Elements of Ontology

In this part of the User Manual we will observe some basic steps to create an ontology by ArchiGraph.Mir ontology editor. You will learn how to build your own model of a part of the world[1]. How big the part will be it depends only on you.

Creating and Editing Classes

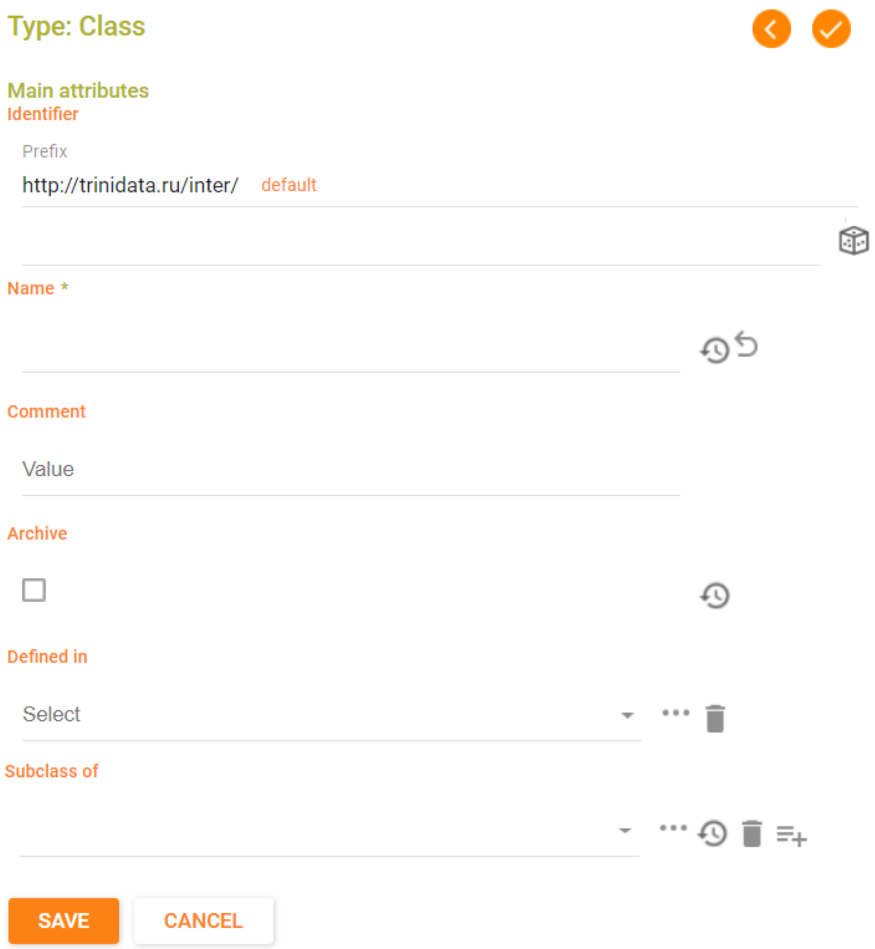
For user's convenience the classes in ArchiGraph.Mir are presented in the form of a hierarchy built on the "is a subclass" (`subClassOf`) relationship. This hierarchy is not normalized. For example, one class can be a subclass of several classes at once.

After entering the newly created endpoint[2] there are no classes in the tree on the left. To create a new class, you need to click + on the toolbar.



A form for creating a class will appear on the right:

[1] "Mir" means "world" in Russian.
[2] See ArchiGraph.MDM Administrator's Guide for endpoints management description.



Type: Class

Main attributes

Identifier

Prefix

http://trinidata.ru/inter/ default

Name *

Comment

Value

Archive

□

Defined in

Select

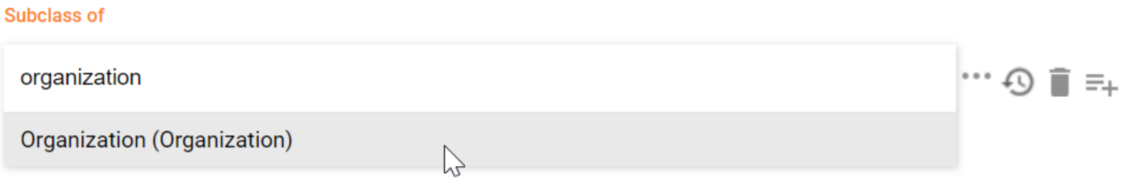
Subclass of

Enter the word “Organization” in “Name” field. When you click “Save” button the word will be copied into “Identifier” field. Spaces and special characters will be removed if any were present in the entered text. ID is the trailing part of the unique identifier of an object (URI). It should not contain spaces or special characters so they will be filtered when saving a record. You can automatically generate a unique identifier pressing 🎲 to the right of the input line.

The non-unique part of the identifier is indicated in the “Prefix” field. The name is the readable name of the entity (`rdfs:label`) that will be used when it is selected in the editor by user.

After clicking the Save button, the Organization class will appear in the tree on the left and will be automatically selected.

Now we need to create the "Company" class as its subclass. You need to click ➕ on the toolbar again to do this. The form on the right side of the page will be cleared again, and it will be possible to enter "Company" as a name for the new class. Let's pay attention to the bottom of the form:



Subclass of

organization

Organization (Organization)

The “Subclass of” field specifies the parent class for the generated class. By default, the class selected in the tree on the left is selected here.

You can clear the value in the selected field by deleting the text from the “Subclass of” field or by clicking the button 🗑 .

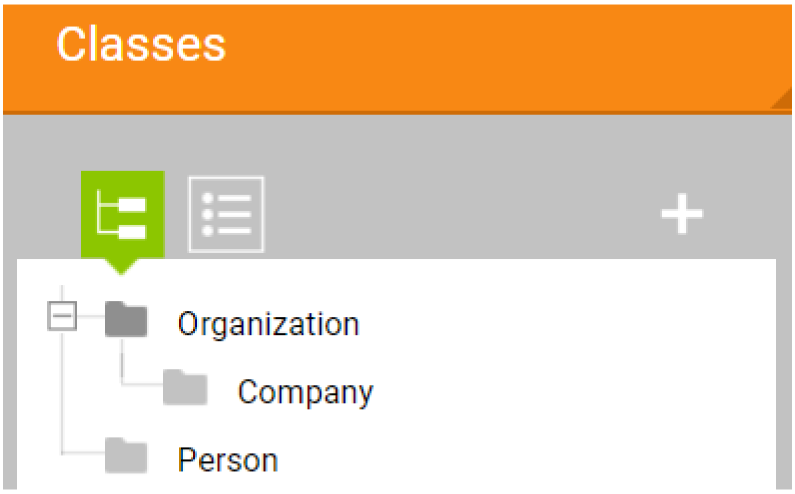
Button ••• opens a dialog box for selecting an object where you can select any other superclass for the created class. In this dialog box, you can quickly find the desired element by selection by part of the name. Clicking on an item name selects it.

You can also start typing the name of the item to be selected in the field instead of opening the dialog box. Then select the item you want from the prompts that appear.

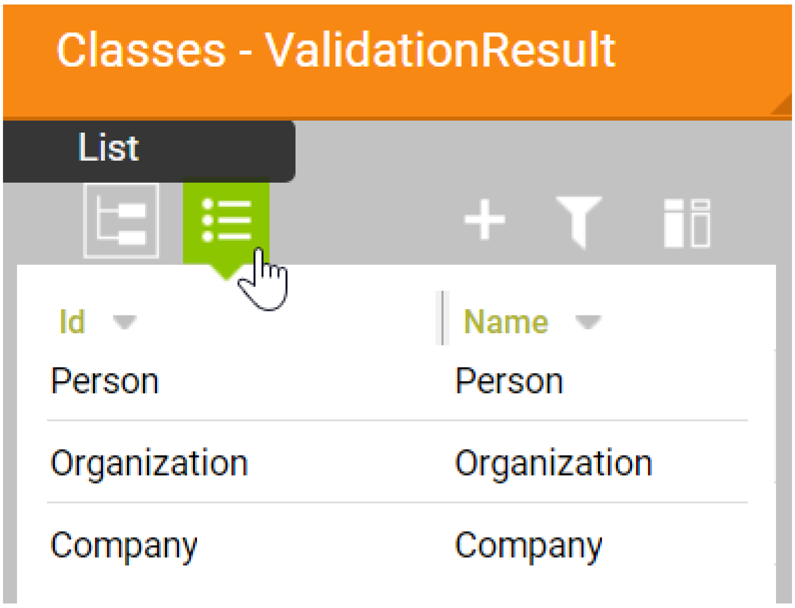
Using the same tools, you can transfer an already created class from one branch of the ontology to another (change its superclass) or add a new superclass in addition to the existing one.

After the class is created, the button ≡+ will appear next to the “Subclass of” field on the right side of the page. By pressing it you can add additional superclasses for current class.

After creating three classes from the above Sample ontology the class tree will be as following:



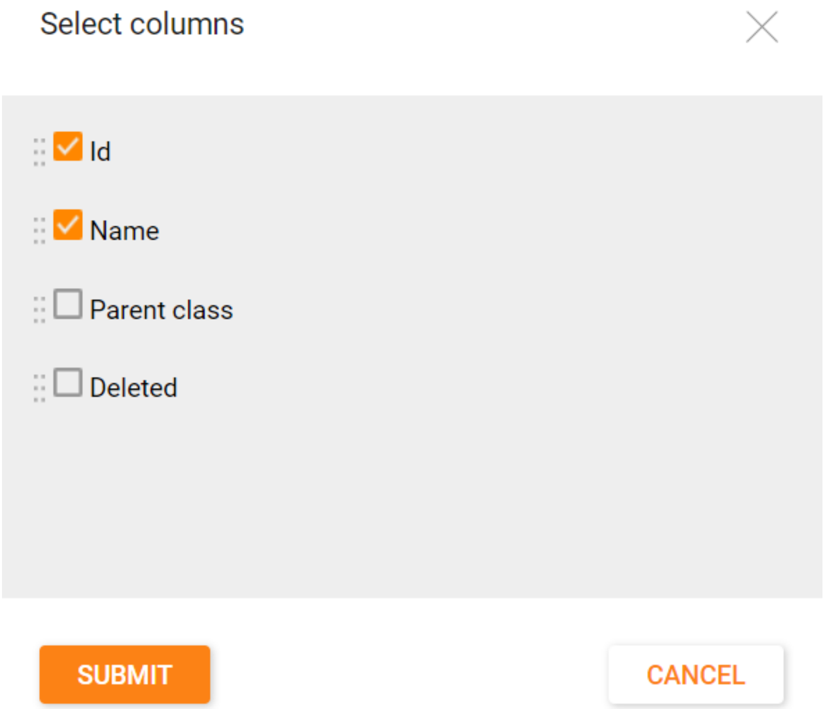
You can switch to the view of classes in the form of a list by selecting the second radio button “List” above the class tree:



Id	Name
Person	Person
Organization	Organization
Company	Company

In this mode class filtering is available, and it is also possible to add additional columns to the list for displaying class properties by clicking the button ◫ .

After clicking on it a dialog box is displayed in which you can use the checkboxes to select the class properties that you want to display in the list:



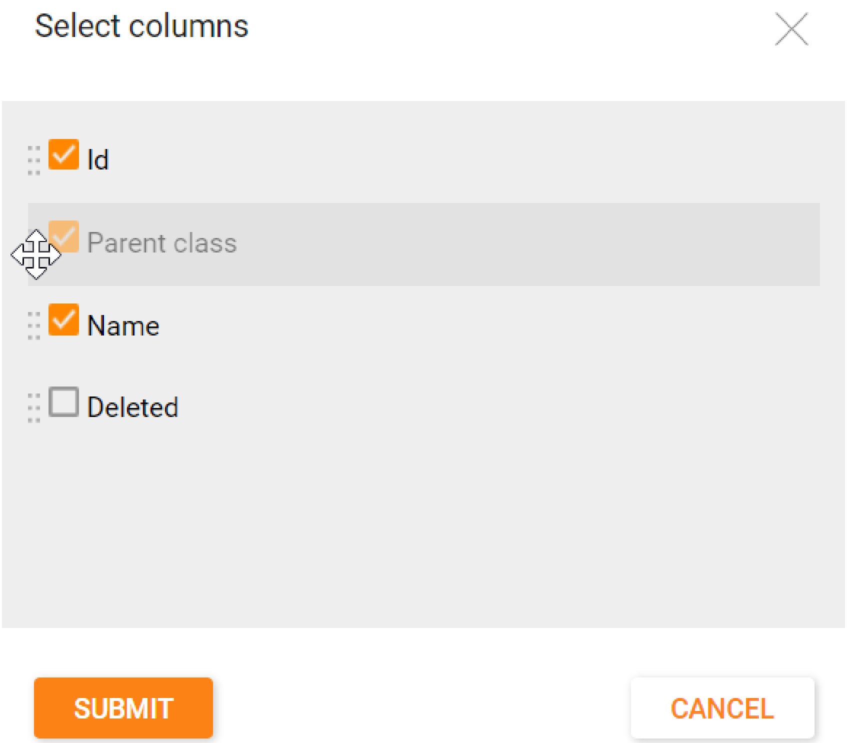
☑ Id

☑ Name

□ Parent class

□ Deleted

You can change the order of the selected columns by drag-and-drop:



☑ Id

☑ Parent class

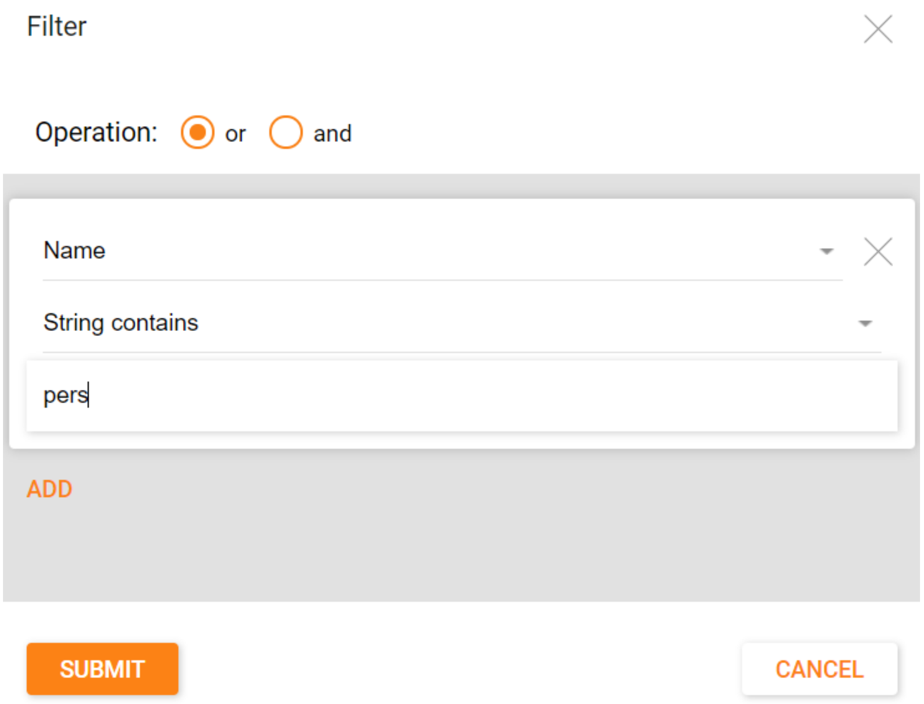
☑ Name

□ Deleted

Sorting and Searching Classes

In list mode, the functions of sorting and filtering the list of classes are available. Sorting option is indicated by a triangle in column. Sorting is set by a single click the name of the column. When you click the column’s name again, the sort changes its direction. Sorting can be reset by pressing the button 🧹.

Classes are filtered using a dialog box that opens when you click the button ▼ above the list. In the window, you can create one or several selection conditions using the "Add" button:



Filter ×

Operation: ● or ○ and

Name ▾ ×

String contains ▾

pers

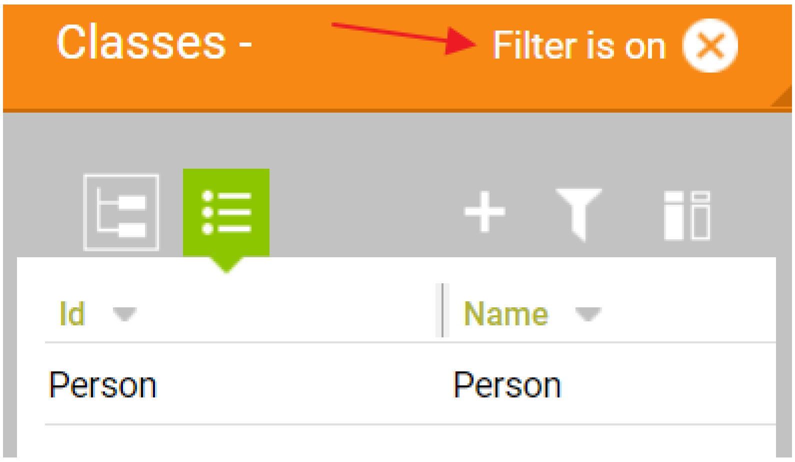
ADD

SUBMIT CANCEL

Each filter condition includes an attribute, condition type and value.

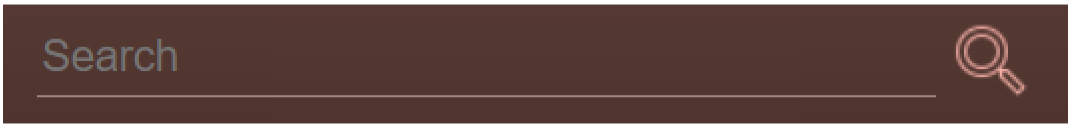
By clicking the "Add" button again you can create additional conditions. All conditions are combined with each other by "AND" or "OR" operators which are selected using the switch above the list of conditions.

The "Submit" button closes the dialog box and filters the list. Only objects that meet the filter condition remain in it. Information that the filter is activated is displayed above the list:



Pressing the filter button again allows you to edit the set conditions.

You can find any element of the ontology other than individual objects using the search bar located at the top of the page.

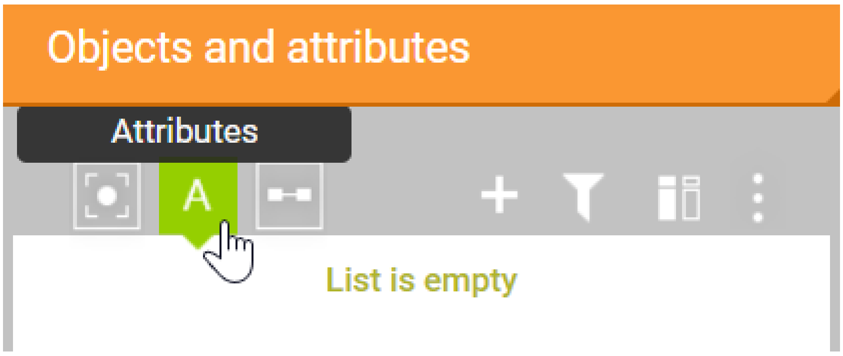


Creating Datatype Properties

Any ontology contains information about something. To be able to express meaningful information about ontology objects, it is necessary to create sets of properties for them. Properties are divided into two types: some accept literal values (strings, numbers, etc.), others are designed to bind different objects together.

Let's consider creating a datatype (literal) property. You need to select a class in the tree the objects of which will have the property being created.

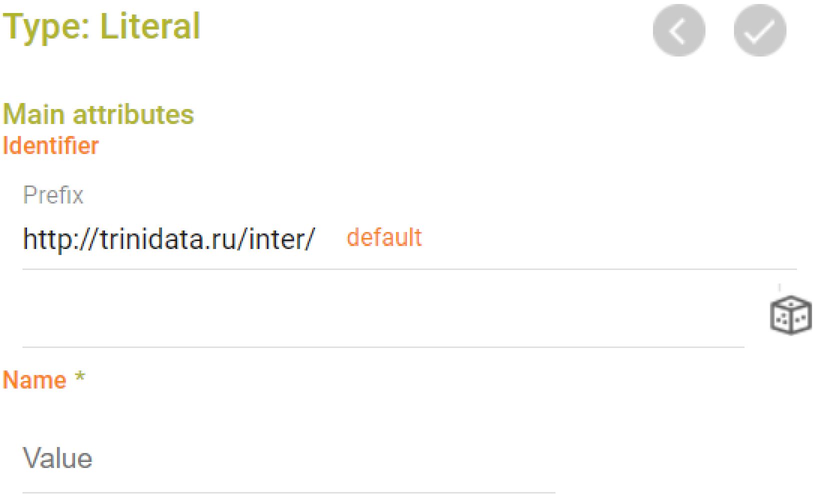
Switch to the list of literal properties by selecting “Attributes” button above the list of entities in the “Objects and attributes section”:



Objects and attributes

List is empty

Clicking the + button in the toolbar above this list, you should make sure that in the title of the entity creation form its type is displayed as “Literal”:



Type: Literal

Main attributes

Identifier

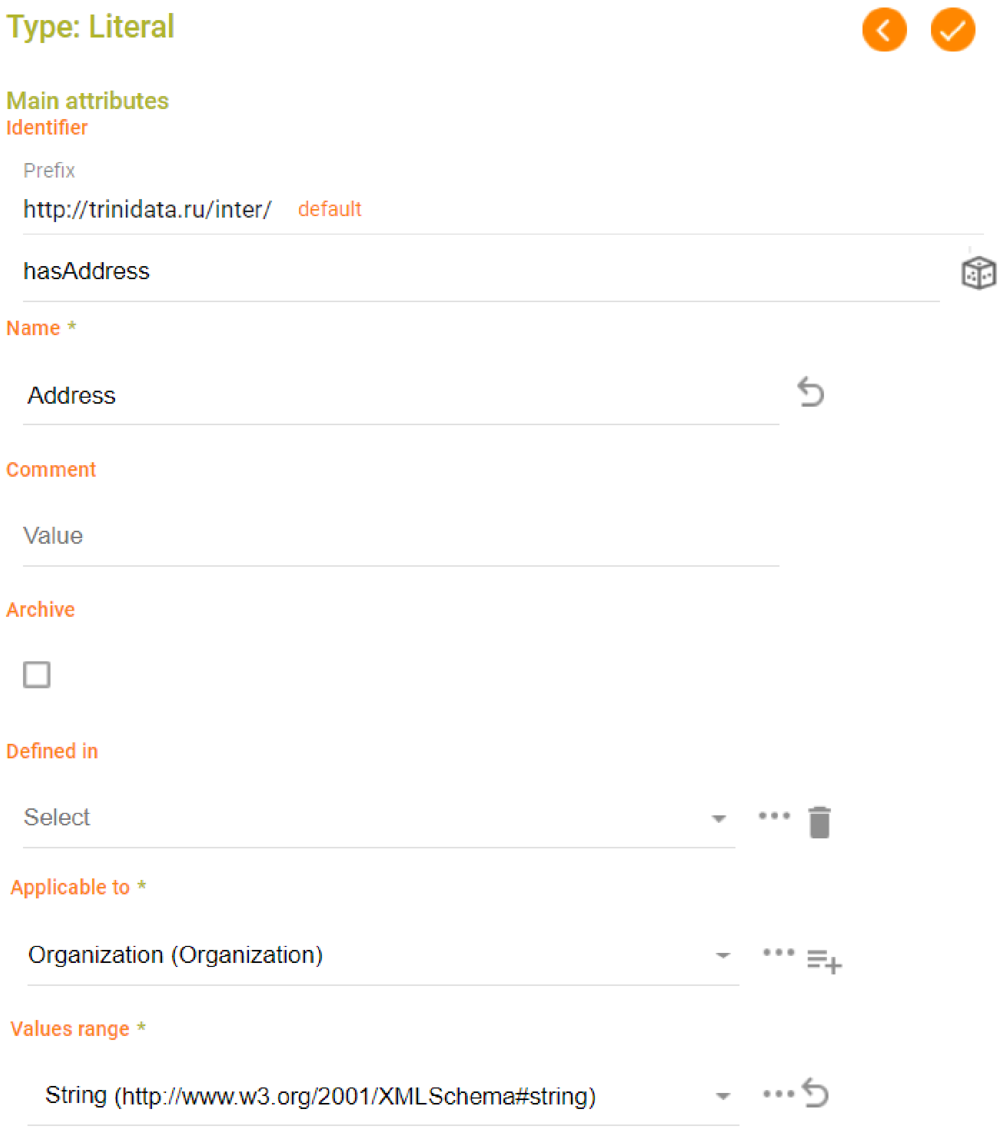
Prefix

http://trinidata.ru/inter/ default

Name *

Value

According to our Sample Ontology it is necessary to create the `hasAddress` literal property. The form for creating it is like the form for creating a class, but it differs in the “Applicable to” and “Values range” fields.



Type: Literal

Main attributes

Identifier

Prefix

http://trinidata.ru/inter/ default

hasAddress

Name *

Address

Comment

Value

Archive

□

Defined in

Select

Applicable to *

Organization (Organization)

Values range *

String (http://www.w3.org/2001/XMLSchema#string)

The value in the “Values range” field is selected using the dialog box which displays a list of XSD data types supported by ArchiGraph Platform.

In the “Applicable to” field you can specify not one but several classes. In this case their set is considered as a union, which means that the property will be applicable to objects of any of the specified classes.

The ArchiGraph.Mir administrator can manage the option to select several classes for “Applicable to” function. This setting is usually performed during system deployment and is not changed later. For the following examples in User Manual, it is assumed that this constant is set to “Yes”, so the property will apply to all objects that belong to at least one applicable class. If the administrator had set this constant to "No", the properties would only apply to objects that belong to all of the listed classes.

After the property `hasAddress` is created you can specify that it is inherent not only to objects of the “Company” class but also to “Person” class. To do this, click the ≡+ button on the right of saved value in “Applicable to” field. A second line appears for the next instance of the attribute value. It is necessary to click the value selection button ••• , and in the dialog box that appears select (or find by name) the “Person” class.

After that you need to save the form. Then you can add another instance of the property value, and so on.

To make sure that a property is created that is inherent to objects of two different classes you need to select the "Organization" class in the tree. Then click “Attributes” selector in “Object and attributes” section and the property should be displayed in the list.

Selecting the "Person" class in the tree you can make sure that the `hasAddress` property is applicable to its objects as well.

Following our Sample Ontology you need to create the second property `hasBirthdate`. Do this by analogy with creating an “Address” property. The differences are that you need to select "Date" in the “Values range” field and leave only "Person" class in the “Applicable to” field.

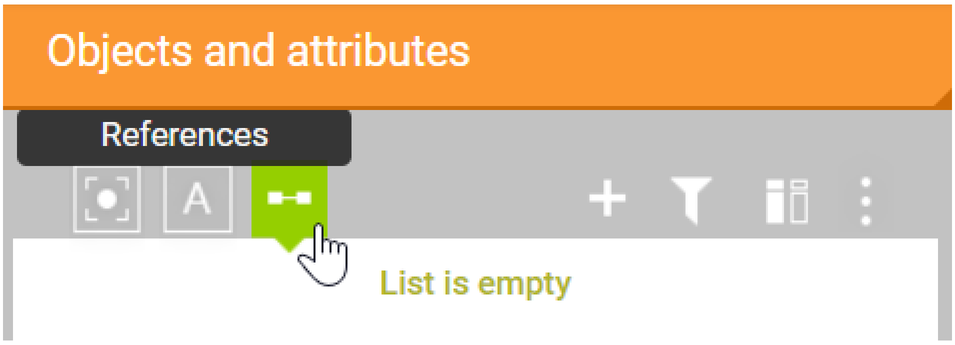
Since each person has one and only one date of birth the “Min cardinality” and “Max cardinality” fields are to be set to ‘1’. These fields indicate how many values a given property can have for each individual object.

The ability to have more than one value for each property is one of the differences between the semantic information model and the relational one. If the minimum number of values (owl: minCardinality) is set to 1 this means that for each individual object of those classes to which the property is applicable, you need to specify its value (the property is mandatory).

If the maximum number of values (owl: maxCardinality) is set to 1 this means that a property cannot have more than one value set for each individual object.

Creating Object Properties

Object property (reference) is created in the same way as datatype (literal) property. To create an object property you need to select a class, then switch to the list of object properties by selecting “References” button above the list of entities in the “Objects and attributes section”:



Accordingly to our Sample Ontology you need to create the property `isFounder`. Select the “Person” class in the tree whose individual objects will have this property, click “References” button in the “Objects and attributes section” and then + . The property creation form will appear.

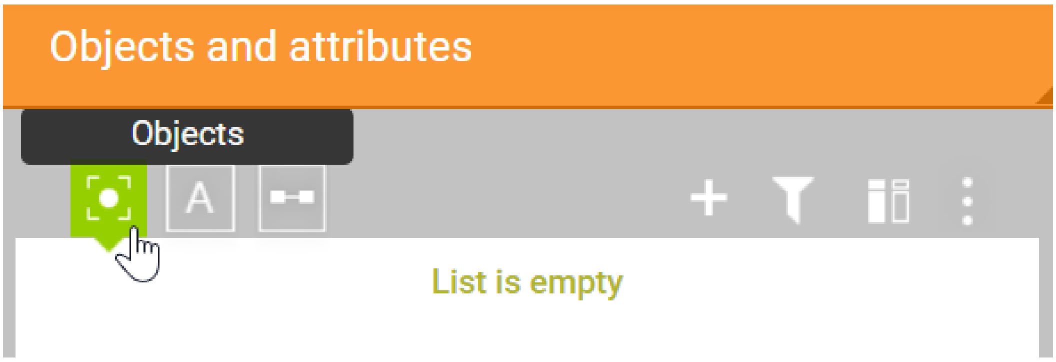
As with literal properties, multiple classes can be specified in the “Applicable to” field.

The “Values range” field is turned into select-an-object tool when creating the reference. By this tool you need to select a class instances of which this property will refer to. Moreover, for object properties you can specify several classes for the "Values range" field. In this case the value of the object property can be an object of any of the specified classes.

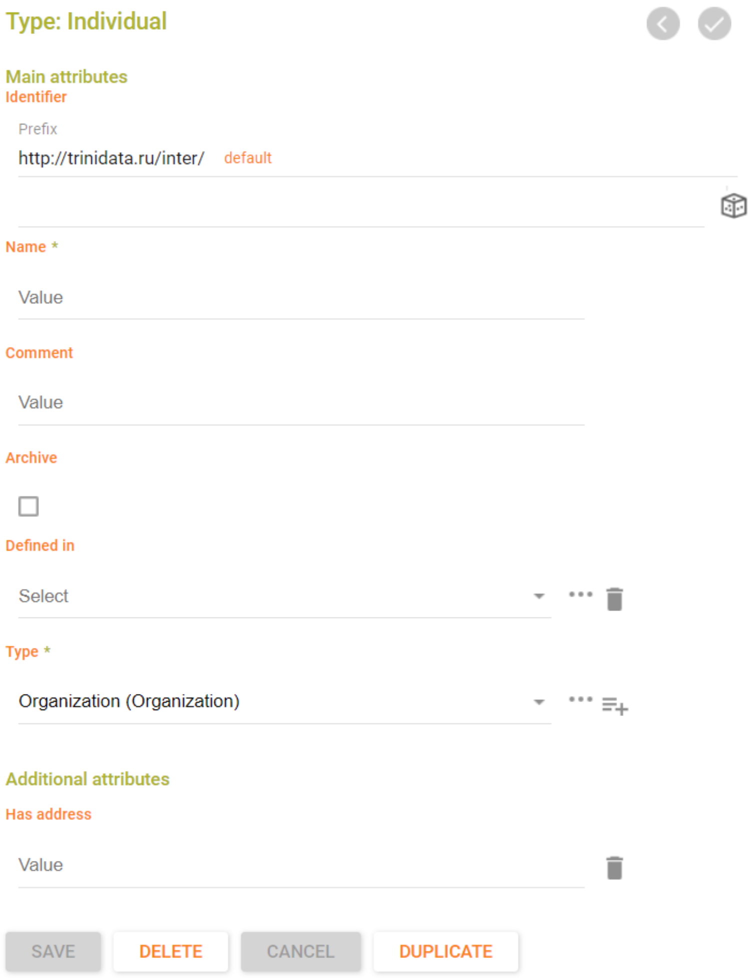
Working with property lists is the same as with a list of classes. You can set a filter, select columns, apply sorting by clicking on the header of column.

Creating Individual Objects

Individual objects of ontology reflect information about specific objects of the real world related to previously created classes. To create an individual object select the parent class in the tree. Then switch to the list of objects clicking “Objects” button above the list of entities in the “Objects and attributes section”:



Click + button above the list, the individual object creation form will be displayed on the right:



Type: Individual

Main attributes

Identifier

Prefix

http://trinidata.ru/inter/ default

Name *

Value

Comment

Value

Archive

□

Defined in

Select

Type *

Organization (Organization)

Additional attributes

Has address

Value

Please note that the entity type "Individual" must be indicated above the list.

The identifier does not need to be specified for individual objectsю It will be generated automatically from the class name and a unique random number. Clicking on the button 🎲 generates a random object identifier in GUID format.

After clicking the “Save” button, an individual object will be created, and it will be displayed in the list of objects.

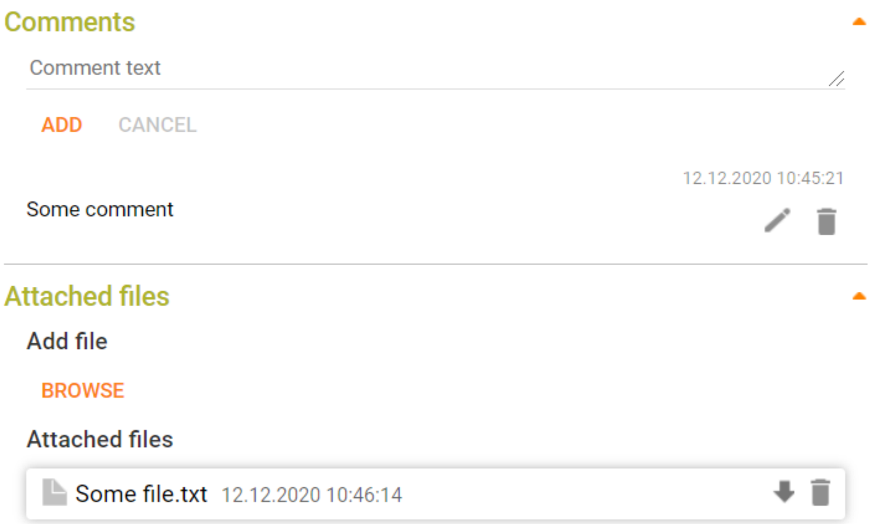
As an individual object is selected in the list you can edit all its properties. Pay attention to the fact that the object has a set of properties defined for the class to which it belongs, as well as inherited from parent classes.

Advanced Topics

In this part some additional features of ArchiGraph.Mir ontology editor are observed. These features are quite useful for managing the information your ontology contains.

Comments and Attaching the Files

Any user who has access to the ontology object can leave a comment or attach a file. At the bottom of the individual object’s section there are tools for comments and files:



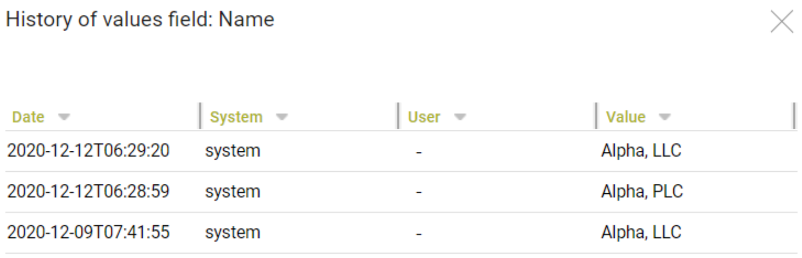
12.12.2020 10:45:21
Some comment

Some file.txt 12.12.2020 10:46:14

Use 🗑 to delete a comment or file or ✎ to edit file comment. By clicking the file name you can downloads it.

Changes and History Browsing

In case the value of any attribute has changed, the editor allows you to view the history of changes. To browse the history use a history view button 🕘 which appears to the right of the field. When you click it the value history window appears:



History of values field: Name

Date	System	User	Value
2020-12-12T06:29:20	system	-	Alpha, LLC
2020-12-12T06:28:59	system	-	Alpha, PLC
2020-12-09T07:41:55	system	-	Alpha, LLC

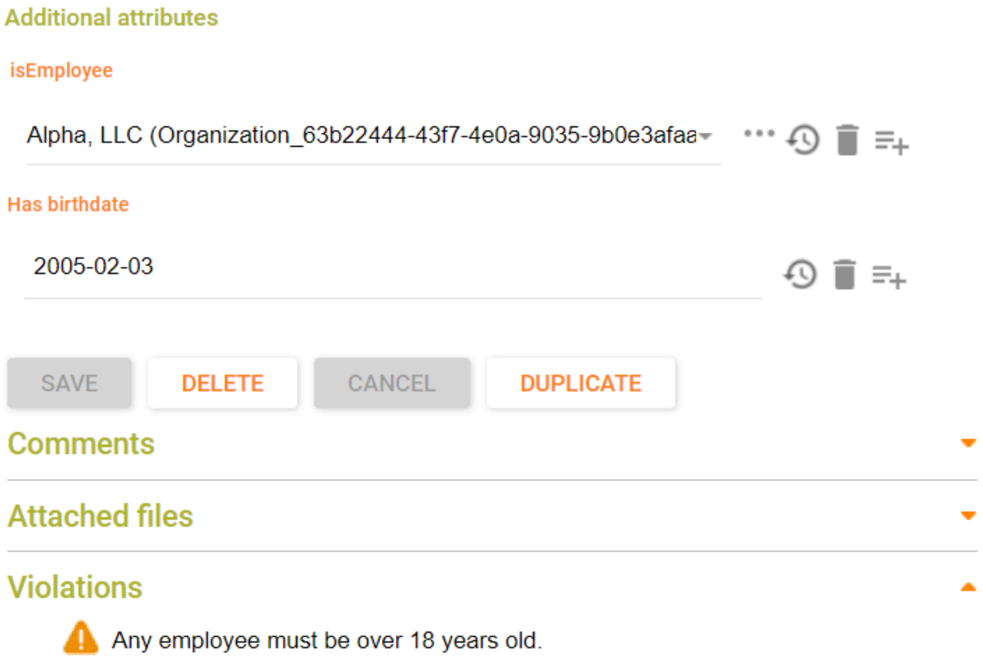
Here you see all the values the object's attribute ever had. Each row in the table corresponds to one change. The "System" column displays the identifier of the automated system i.e. the client of ArchiGraph Platform on whose behalf the operation to change the object properties was performed. If the automated system sends the username it will be displayed in the "User" column. The "Value" column indicates the value of the property that was assigned as a result of the operation. If the property has multiple values they are separated by commas.

You cannot delete history or manage it in any way, even with administrator rights. The history of values is stored not in the ontological model itself but in the metadata repository.

Violations Indicators Panel

The ArhiGraph Platform includes ArchiGraph.KMS application where special rules of consistency checking can be configured.

Any violations of rules set in ArchiGraph.KMS application will be displayed on the "Violations" panel at the bottom of individual object's section:



Additional attributes

isEmployee

Alpha, LLC (Organization_63b22444-43f7-4e0a-9035-9b0e3afaa

Has birthdate

2005-02-03

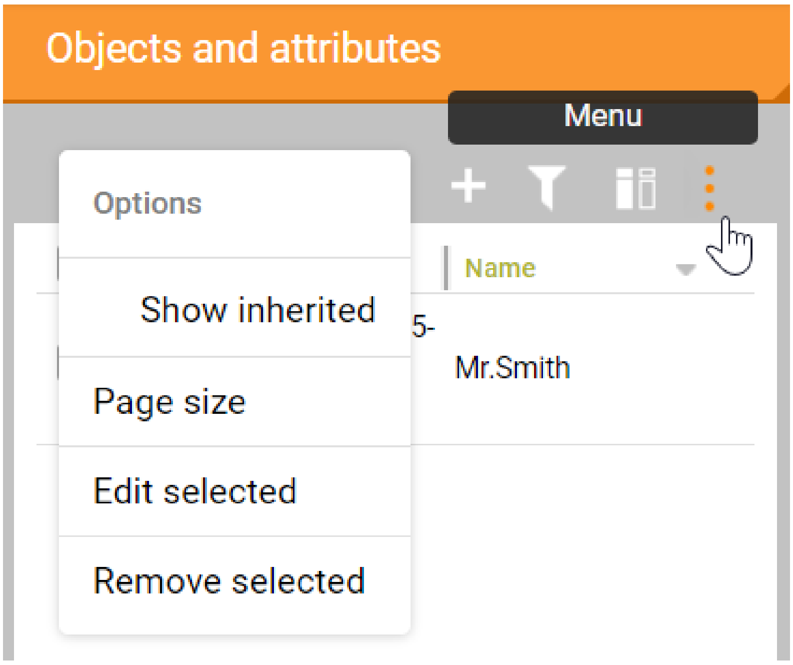
Any employee must be over 18 years old.

Operations with Groups of Individual Objects

A number of group operations are available for individual objects in ArchGraph.Mir ontology editor:

- Remove multiple objects.
- Assigning a specific value to any property of a group of individual objects.

These actions are called using the menu button ⋮ above the list of objects:

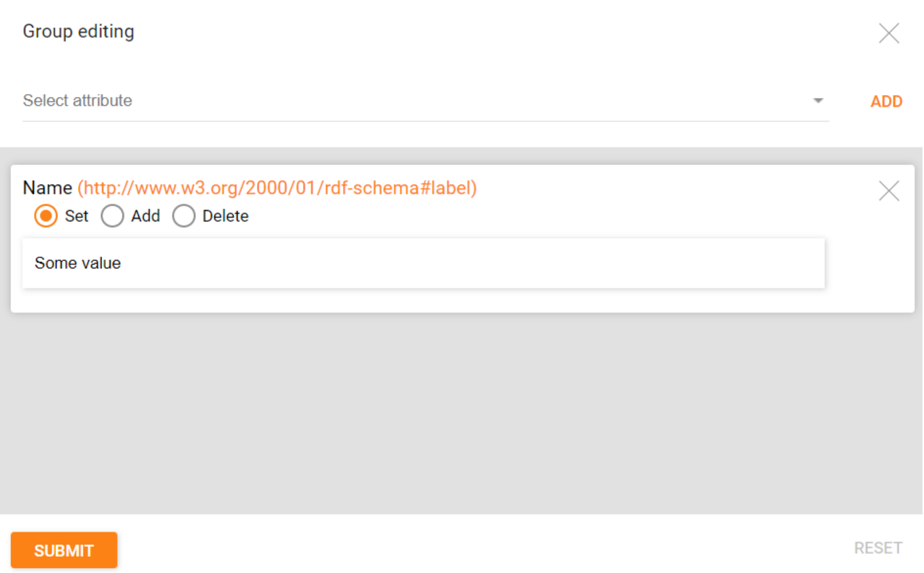


Using the “Page Size” option you can select the number of objects to display on the page.

Before applying “Edit selected” and “Remove selected” options you need to select the objects with checkboxes on which you want to perform the operation. It can be done for all objects at once by pressing the checkbox button next to the header of the first column of the list of objects. Clicking on it again will invert the selection. When you select all the objects you need and choose the “Remove selected” option the editor asks for confirmation and if you confirm all marked entries will be deleted.

ATTENTION
You cannot undo this operation!

When you choose "Edit Selected" option the "Group Editing" dialog box will appear. You need to select a property and enter or select its value:



Group editing

Select attribute ADD

Name (http://www.w3.org/2000/01/rdf-schema#label)

● Set ○ Add ○ Delete

Some value

SUBMIT RESET

After clicking the "Submit" button the specified value will be assigned to all individuals objects marked in the list.

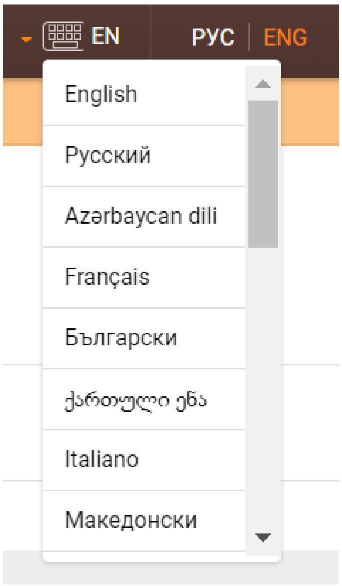
The "Add" switch allows you to add a new property value without erasing the existing ones.

With this function it is possible to move several objects from one class to another. To do this select the "Type" property in the window and select the receiver class.

Content and GUI Languages

ArchiGraph Platform allows storing versions of object property values in different languages.

To switch the content input language use a switch located in the upper panel of the editor:



After a language is selected the editor's forms and lists will display property values in the appropriate language, if any.

If the value of any property in the selected language is not set then the field with its content will be empty.

You can also change the interface language (GUI) by the buttons next to the content language one. You can switch the language of section names and standard attributes. English and Russian are supported.

Team Working

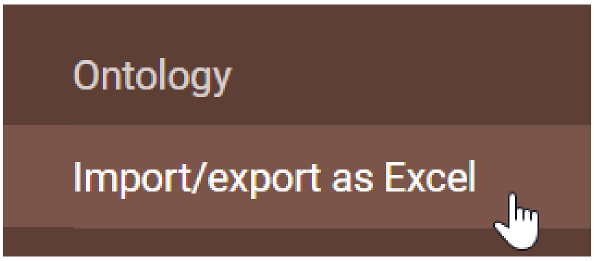
ArchiGraph.Mir editor is designed for simultaneous work of several users. Any change made by one of the users to any element of the ontology is sent on to other instances of the editor open in the browsers of other users. These changes are displayed to other users immediately without reloading the page.

In case the user edits the property of an object in a form and at the same time another user has made changes to this object and saved it so the other users who are editing this object will no longer be able to save the changes.

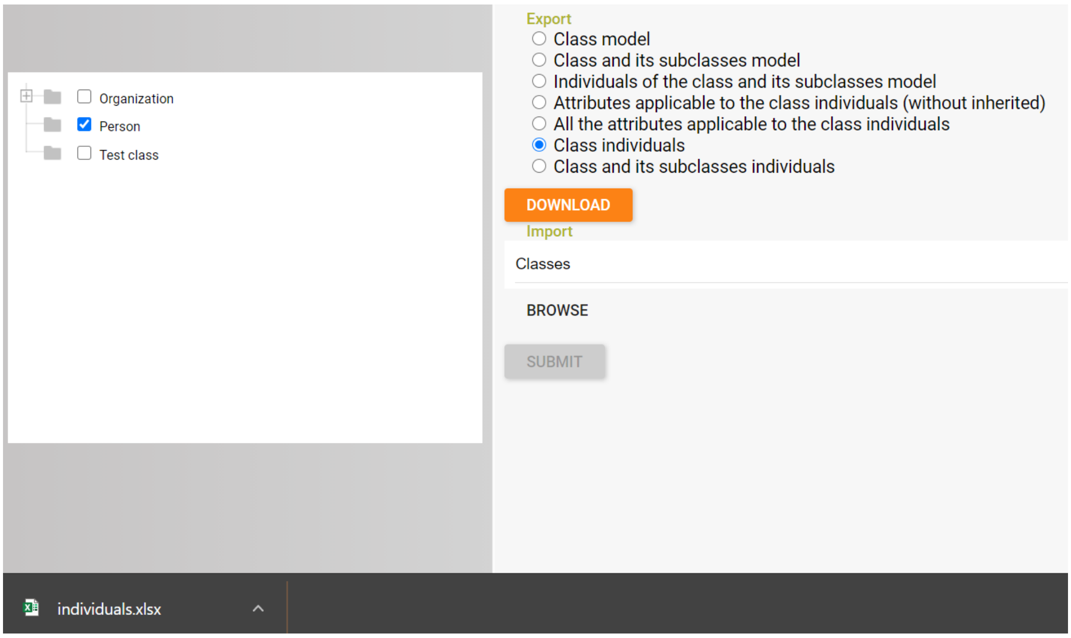
They need to select the object in the list again to load the current set of values of its properties and continue editing. They need to take the new current version of the object as the initial state.

Export and Import

The ArchiGraph.Mir editor allows downloading ontology elements into MS Excel file and loading data composed in MS Excel file back into the ontology. To do this, click the menu button in the upper left corner and select “Import/export as Excel”:



The data download / upload interface looks like this:



☐ Organization
☑ Person
☐ Test class

Export

○ Class model
○ Class and its subclasses model
○ Individuals of the class and its subclasses model
○ Attributes applicable to the class individuals (without inherited)
○ All the attributes applicable to the class individuals
● Class individuals
○ Class and its subclasses individuals

Import

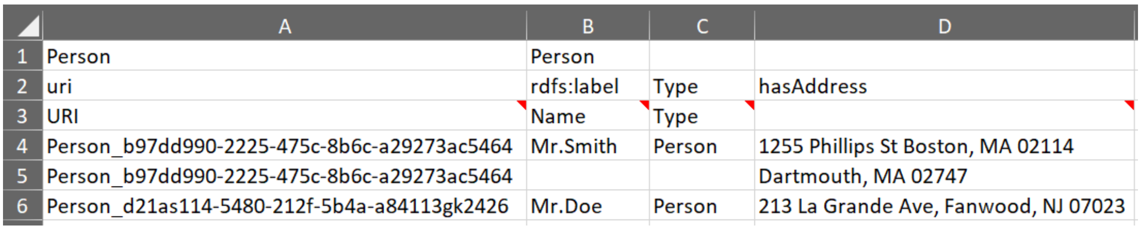
Classes

To download ontology elements you need to select one or more classes in the tree on the left with checkboxes. Then you need to select one of the following options in the central part of the page:

- “Class and its subclasses model” downloads a MS Excel sheet where you can fill in values for objects of selected classes and their subclasses.
- “Class individuals” downloads all individual objects of those classes that are selected in the tree.
- “Class and its subclasses individuals” downloads the same as in the previous paragraph, but the MS Excel sheet includes all individual objects of subclasses of the selected classes.
- “Class model” downloads the list of classes and the values of their properties.
- “Class and its subclasses model” downloads the same as in the previous paragraph but the MS Excel sheet also contains the descriptions of all subclasses of the selected classes.
- “Attributes applicable to the class individuals (without inherited)” downloads a list of attributes whose “Applicable to” (rdfs:domain) corresponds to the selected classes.
- “All the attributes applicable to the class individuals” downloads the list of attributes applicable to individual objects of the selected classes including the results of inheriting attributes from superclasses.

All downloaded MS Excel files have a similar structure: rows represent ontology elements, columns correspond to their attributes, cells contain attribute values for specific elements.

The first column always contains the object identifier. If any attribute of the object has more than one value, then you need to create an additional line for this element which will contain the same identifier as in the main line. You need to fill in only those cell that correspond to attributes that have more than one value:



	A	B	C	D
1	Person	Person		
2	uri	rdfs:label	Type	hasAddress
3	URI	Name	Type	
4	Person_b97dd990-2225-475c-8b6c-a29273ac5464	Mr.Smith	Person	1255 Phillips St Boston, MA 02114
5	Person_b97dd990-2225-475c-8b6c-a29273ac5464			Dartmouth, MA 02747
6	Person_d21as114-5480-212f-5b4a-a84113gk2426	Mr.Doe	Person	213 La Grande Ave, Fanwood, NJ 07023

In this example lines 4 and 5 correspond to the same individual object with the identifier `Person_b97dd990-2225-475c-8b6c-a29273ac5464` which has two values for the `hasAddress` attribute (“1255 Phillips St Boston, MA 02114” and “Dartmouth, MA 02747”).

Downloaded MS Excel file can be saved on the user's computer, edited and uploaded back into the editor. To do this choose the type of the data you want to upload in “Import” section (Classes, Attributes or Individual Objects), then click “Browse” button, choose your MS Excel file and press “Submit”.

Those uploaded objects whose identifiers are absent in the ontology will be created, and existing objects will be edited. No changes will be made with those objects that exist in the ontology but are absent in the file. When the file is imported messages about the results of the operations performed will be displayed.

It is important to keep the column names and their structure in accordance with the downloaded elements of the ontology, otherwise the imported elements will not be included in the model, and the editor will report an error.

We recommend you download the necessary elements from the editor firstly to arrange new data on MS Excel sheet in the same structure.