

# Technologies of Semantic WEB as an environment of application development and integration

**Evgeny Cherkashin, Nikita Dorodnykh, Alexey Shigarov, *et al.***

Matrosov Institute for System Dynamics and Control Theory of  
Siberian Branch of Russian Academy of Sciences, Irkutsk, Russia

[eugeneai@icc.ru](mailto:eugeneai@icc.ru)

AICTS'2021, December, 06, 2021  
Irkutsk, Russia

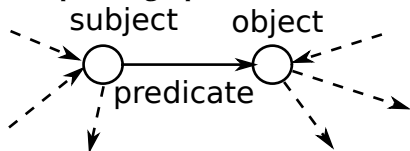
**Main objective** of the activity is to construct data integration tools based on the **standardized** Semantic WEB technologies.

The following aspects are under consideration:

1. Application data representation
2. Ontological model representation
3. Document publication
4. Application integration
5. Model transformation

# Representation of ontological models

The ontologies are represented with  $\langle \text{subject}, \text{predicate}, \text{object} \rangle$  **triples** as **graphs**, and there is frequently a **context**, the graph itself.

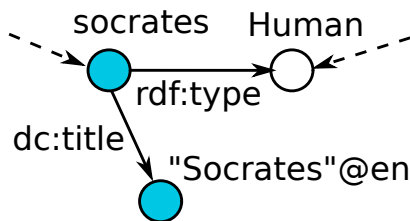


The subjects, the predicates and *some* objects are **URI/IRI**. E.g., <http://purl.org/dc/terms/> defines the **namespace** “dc”.

*Other* subjects are **literals**.

All **XML** properties are applicable.

- ❑ XML format for data representation (optional!)
- ❑ global identification
- ❑ different specification usage in one document



● A-Box

○ T-Box



# Data formats for graph representation

## □ N-Triples

```
<http://mythology.Greek.org/#Cronus>
  <http://www.example.org/schemas/relationship/fatherOf>
    <http://mythology.Greek.org/#Zeus>.
```

## □ Turtle

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix ex: <http://example.org/stuff/1.0/> .
<http://www.w3.org/TR/rdf-syntax-grammar>
  dc:title "RDF/XML Syntax Specification (Revised)" ;
  ex:editor [
    ex:fullname "Dave Beckett";
    ex:homePage <http://purl.org/net/dajobe/>
  ] .
```

## □ Notation 3 (N3)

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
<http://en.wikipedia.org/wiki/Tony_Benn>
  dc:title "Tony Benn" ;
  dc:publisher "Wikipedia" .
```

## □ RDF/XML

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/
    22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://en.wikipedia.org/
    wiki/Tony_Benn">
    <dc:title>Tony Benn</dc:title>
    <dc:publisher>Wikipedia</dc:publisher>
  </rdf:Description>
</rdf:RDF>
```

## □ JSON-LD

```
{
  "@context": {
    "name": "http://xmlns.com/foaf/0.1/name",
    "homepage": {
      "@id": "http://xmlns.com/foaf/0.1/
        workplaceHomepage",
      "@type": "@id"
    },
    "Person": "http://xmlns.com/foaf/0.1/Person"
  },
  "@id": "http://me.markus-lanthaler.com",
  "@type": "Person",
  "name": "Markus Lanthaler",
  "homepage": "http://www.tugraz.at/"
}
```

# Resource storage and access

Semantic WEB documents are stored as **files**, **documents**, and, in general, [cloud] **resources** on servers.

Popular server software are

- ❑ Openlink Virtuoso (DBPedia.org)
- ❑ Apache Jena (also a Java library)
- ❑ GraphDB (has good control interface)
- ❑ ClioPatria (not so popular, has integrated Prolog engine)

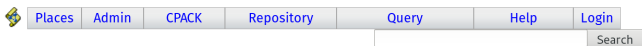
Further info is at

<https://www.w3.org/wiki/SparqlImplementations>.

**SPARQL** is a language to formulate questions (queries) for knowledge databases

```
SELECT ?publisher ?publisherLabel (AVG(?pages) AS ?avgPages)
WHERE
{
  ?book wdt:P123 ?publisher;
        wdt:P1104 ?pages.
  SERVICE wikibase:label { bd:serviceParam
    wikibase:language "[AUTO_LANGUAGE]". }
}
GROUP BY ?publisher ?publisherLabel
HAVING (COUNT(?book) > 1)
ORDER BY DESC(?avgPages)
```

# Ontological instruments: ClioPatria



## Local view for "<http://cliopatria.swi-prolog.org/schema/cpack#gitURL>"

Predicate	Value (sorted: <a href="#">default</a> )
<a href="#">rdfs:label</a>	"GIT URL"
<a href="#">rdf:type</a>	<a href="#">owl:ObjectProperty</a>
<a href="#">rdfs:domain</a>	<a href="#">cpack:Software repository</a>
<a href="#">rdfs:range</a>	<a href="#">rdfs:Resource</a>
<a href="#">rdfs:comment</a>	"URL to clone the repository using git"
<a href="#">rdfs:subPropertyOf</a>	<a href="#">cpack:SCM URL</a>

All properties reside in the graph <http://cliopatria.swi-prolog.org/schema/cpack>

The resource does not appear as an object

## Predicate statistics

Predicate	#Triples	#Distinct subjects	#Distinct objects	Domain(s)	Range(s)
<a href="#">cpack:gitURL</a>	<a href="#">102</a>	<a href="#">102</a>	<a href="#">99</a>	<a href="#">2</a>	<a href="#">rdfs:Resource</a>

## Context graph



Using standard vocabularies form cross-application platform, *e.g.*, interpreting relations.

# Semantic web technologies & Knowledge graphs

Semantic Web (WEB 3.0) is characterized with

- ❑ Technological basis, oriented to the web
- ❑ Standardized data formats, storage, and processing
- ❑ Open principles of data publishing
- ❑ Services for data storage and access provision
- ❑ Generalized and special user interfaces are used for data presentation

For the Knowledge Graphs (KG), the following is of interest.

- ❑ Converged notions **data** and **knowledge** as something is **known**
- ❑ Contain data, relations, and metadata (vocabularies)
- ❑ Distinguished **node filling in** and **processing** graph triples, *e.g.*, with SPARQL queries with UPDATES
- ❑ Allow **postpone** the formal definition of a schema
- ❑ Three types of graph schemata: **semantic** (aimed at generalization), **validating** (*e.g.* semantics, **completeness** w.r.t. sets of relations), and **emergent** (infer a set of generalized structures and **reconstruct** the KG).



# Knowledge graph: Validating semantic example



Fig. 10. Example class hierarchy for Event

Table 2. Definitions for sub-class, sub-property, domain and range features in semantic schemata

Feature	Definition	Condition	Example
SUBCLASS	$c \text{--} \text{subc. of} \rightarrow d$	$x \text{--} \text{type} \rightarrow c \text{ implies } x \text{--} \text{type} \rightarrow d$	City $\text{--} \text{subc. of} \rightarrow$ Place
SUBPROPERTY	$p \text{--} \text{subp. of} \rightarrow q$	$x \text{--} p \rightarrow y \text{ implies } x \text{--} q \rightarrow y$	venue $\text{--} \text{subp. of} \rightarrow$ location
DOMAIN	$p \text{--} \text{domain} \rightarrow c$	$x \text{--} p \rightarrow y \text{ implies } x \text{--} \text{type} \rightarrow c$	venue $\text{--} \text{domain} \rightarrow$ Event
RANGE	$p \text{--} \text{range} \rightarrow c$	$x \text{--} p \rightarrow y \text{ implies } y \text{--} \text{type} \rightarrow c$	venue $\text{--} \text{range} \rightarrow$ Venue

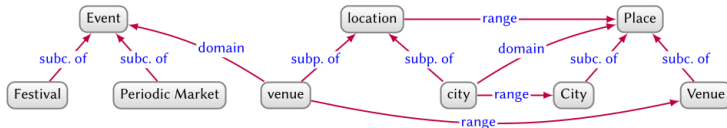
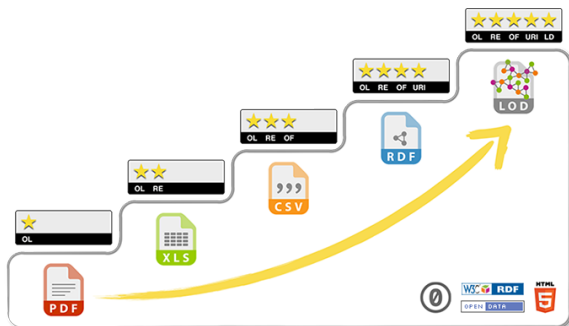


Fig. 11. Example schema graph describing sub-classes, sub-properties, domains, and ranges

# Linked Open Data (LOD) star evaluation

Data are available in

- 1\* any format **openly**
- 2\* a **structured format**, such as Microsoft Excel file format (.xls)
- 3\* a **non-proprietary structured format**, such as .csv
- 4\* **W3C standards**, like using RDF and employing URIs
- 5\* a hypercontent form **having links to other Linked Open Data sources**



# Useful standard vocabularies

## Standardized vocabularies

- ❑ Friend-of-a-friend (**foaf**) for agent information: individuals, legal entities, program agents.
- ❑ Provenance (**prov**) for making references between documents.
- ❑ Dublin Core (**dc**) for published resource metadata mark up.
- ❑ DBPedia resource (**dbr**) to refer external classes and instance objects.
- ❑ Open annotation (**oa**) as an “bookmark” ontology.
- ❑ The Bibliographic Ontology (**bibo**) used for literature reference mark up.
- ❑ Schema.org (**schema**) for Google, Yandex, Yahoo, *etc.* searchable objects, structural elements.

## Non-standard vocabularies

- ❑ Ontology **nssp** for Mothur source code processing results.
- ❑ Ontology **uml** for XML representation.

# Instrumentation: Ontology metadata server LOV

## Linked Open Vocabularies (LOV)

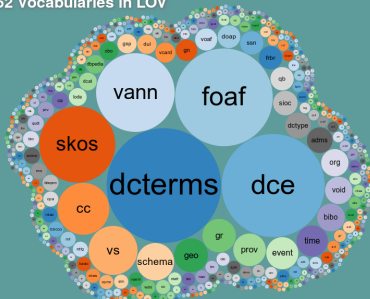
+ Suggest

Documentation

g+ Follow



562 Vocabularies in LOV



### Latest Insertion

**summa** - SUMMA Vocabulary  
2016-08-12

**plink** - PersonLink Ontology  
2016-08-10

**lot-lite** - lot-lite ontology  
2016-07-07

**mdl** - RDF vocabulary to  
describe a Multidimensional  
Interface.  
2016-07-06

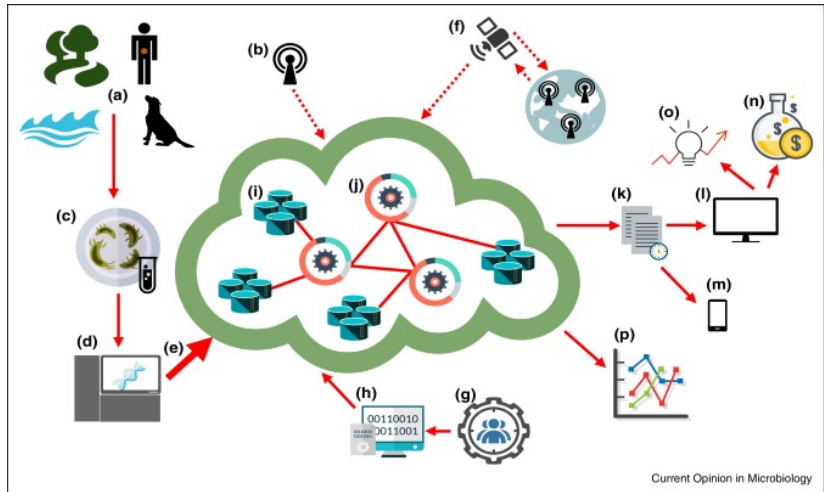
**earth** - The Linked Earth  
Ontology  
2016-06-21

### Latest Updates

**security** - Security Ontology  
2016-08-12

## Applications

# Application: Information infrastructure for supporting Baikal microbiome research



# The aim of the research and development

The object of the research is genetic data processing. We would like to involve biologists in it. The subject is the amplicon data processing with MiSeq SOP<sup>1</sup> (a technique).

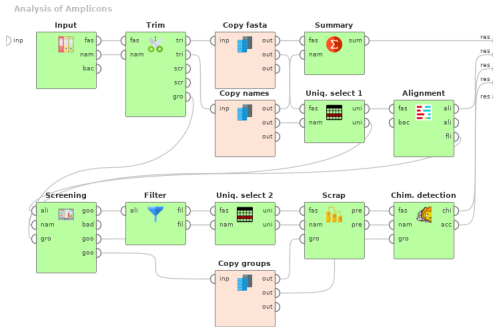
The primary **aim** of the research is to construct infrastructure which comprises

- ❑ Big Data database for sequence storage;
- ❑ metadata storage and adapters;
- ❑ visual construction of a processing model;
- ❑ cloud genetic data processing unit;
- ❑ metadata inference unit;
- ❑ data integration unit based on Semantic Web and Linked Open Data principles.

---

<sup>1</sup>Standard Operational Procedure

# Dataflow representation of NGS analysis of amplicons



Term	Description
NGS	New Generation Sequencing
Amplicon	A DNA or RNA part copied many times
Mothur	A software toolset for NGS research
Rapidminer	A visual tool for data mining modeling and execution

Green blocks are Mothur modules.  
Others are Rapidminer modules.



# Rapidminer module

```
... vector<string> AlignCommand::setParameters(){ // PART OF MODULE SOURCE
try {
    CommandParameter ptemplate("reference", "InputTypes", "", "", "none", "none", "none","",false,true,true); parameters.push_back(ptemplate);
    CommandParameter pcandidate("fasta", "InputTypes", "", "", "none", "none", "none","fasta-alignreport-accnos",false,true,true); parameters.push_back(pcandidate);
    CommandParameter psearch("search", "Multiple", "kmer-blast-suffix", "kmer", "", "", "",false,false,true); parameters.push_back(psearch);
    CommandParameter pksize("ksize", "Number", "", "8", "", "", "",false,false); parameters.push_back(pksize);
    CommandParameter pmatch("match", "Number", "", "1.0", "", "", "",false,false); parameters.push_back(pmatch);
// . . . . .

package com.rapidminer.ngs.operator; // GENERATED JAVA MODULE
// imports

class MothurChimeraCcodeOperator extends MothurGeneratedOperator {
    private InputPort fastaInPort = getInputPorts().createPort("fasta");
    private InputPort referenceInPort = getInputPorts().createPort("reference");
    private OutputPort chimeraOutPort = getOutputPorts().createPort("chimera");
    private OutputPort mapinfoOutPort = getOutputPorts().createPort("mapinfo");
    private OutputPort accnosOutPort = getOutputPorts().createPort("accnos");

    public MothurChimeraCcodeOperator (OperatorDescription description) {
        super(description);
    }
    @Override
    public void doWork() throws OperatorException {
        super();
        // . . . . .
    }
    @Override
    public String getOutputPattern(String type) {
        if (type=="chimera") return "[filename],[tag],ccode.chimeras-[filename],ccode.chimeras";
        if (type=="mapinfo") return "[filename],mapinfo";
        if (type=="accnos") return "[filename],[tag],ccode.accnos-[filename],ccode.accnos";
        return super.getOutputPattern(type);
    }
}
```

```
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
ngsp:spec a ngsp:Specification ;
    ngsp:module mothur:NoCommand,
        mothur:align-check,
        mothur:align-seqs,
# . . . . .
    mothur:align-check a ngsp:Module ;
        ngsp:outputPattern [ a cnt:Chars ;
            ngsp:parameterName "type" ;
            ngsp:pattern [ ngsp:patternString
                "[filename],align.check" ;
                dc:identifier "aligncheck" ] ;
            cnt:chars # . . . . .
# . . . . .
    mothur:align-check-idir-parameter a ngsp:Parameter ;
        ngsp:important false ;
        ngsp:multipleSelectionAllowed false ;
        ngsp:optionsDefault "" ;
        ngsp:required false ;
        ngsp:type mothur:String ;
        dc:title "inputdir" .

    mothur:align-check-map-parameter a ngsp:Parameter ;
        ngsp:important true ;
        ngsp:multipleSelectionAllowed false ;
        ngsp:optionsDefault "" ;
        ngsp:required true ;
        ngsp:type mothur:InputTypes ;
        dc:title "map" .
# . . . . .
```

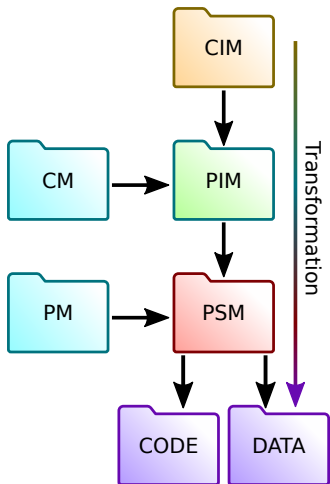
# Procedural data (Mothur tooling of Galaxy)

```
<tool profile="16.07" id="mothur_make_contigs"
name="Make.contigs" version="@WRAPPER_VERSION@.0">
<description>Aligns paired ...</description>
<!-- . . . . -->
<command><![CDATA[ @SHELL_OPTIONS@
## Symlinks creation or On the fly ...
#if input_type.type == 'list_collection'
#for pair in input_type.list_paired_collection:
ln -s {pair.forward} `basename {pair.forward}` &&
ln -s {pair.reverse} `basename {pair.reverse}` &&
echo -e "{pair.name}\t`basename {pair.forward}`\t`
`basename {pair.reverse}`" » combo_fastq.dat &&
#end for ## . . . . .
echo 'make.contigs(
#if input_type.type == 'list_collection':
file=combo_fastq.dat,
#else:
ffastq=ffastq.dat,
rfastq=rfastq.dat,
#end if ## . . . . .
gapextend=gapextend,
rename=rename
processors='{GALAXY_SLOTS:-8}'
)]></command>
<inputs>
<conditional name="input_type">
<param name="type" type="select" label="Select ...">
<option value="regular" selected="true">Two ...</option>
<option value="simple_collection">One pair ...</option>
<option value="list_collection">Multiple ....</option>
</param>
<when value="regular">
<param name="forward_fastq" type="data" />
<param name="reverse_fastq" type="data" />
</when>
</conditional>
<param name="align" type="select" label="..." help="">
```

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
```

```
[ ] a gal:Suite ;
ngsp:module [ a gal:Module,
ngsp:Module ;
gal:command " ## . . . . . ";
gal:exit_code [ gal:level "fatal" ;
gal:range "1:" ] ;
gal:inputs [ gal:checked "false" ;
gal:conditional [ gal:param [ gal:help "" ;
gal:option [ gal:value "yes" ;
dc:description "yes" ],
[ gal:value "no" ;
dc:description "no" ] ;
dc:description "Trim with an oligos file?" ;
dc:title "add" ;
rdfs:range "select" ] ;
gal:when [ gal:value "no" ],
[ gal:param [ gal:min "0" ;
gal:value "0" ;
dc:description "pdiffs - number of differences . . .
dc:title "pdiffs" ;
rdfs:range "integer" ],
[ gal:min "0" ;
gal:value "0" ;
dc:description "bdiffs - number of differences . .
dc:title "bdiffs" ;
rdfs:range "integer" ],
[ gal:min "0" ;
gal:value "0" ;
dc:description "tdiffs - total number of diffe...
dc:title "tdiffs" ;
rdfs:range "integer" ] ] ] ]
dc:identifier "mothur_make_contigs" ;
dc:title "Make.contigs",
"make.contigs" ;
schema:sku 1 ''
```

# Model-Driven Architecture



**CIM** Computationally Independent Model;

**CM** Model of Computations;

**PIM** Platform Independent Model;

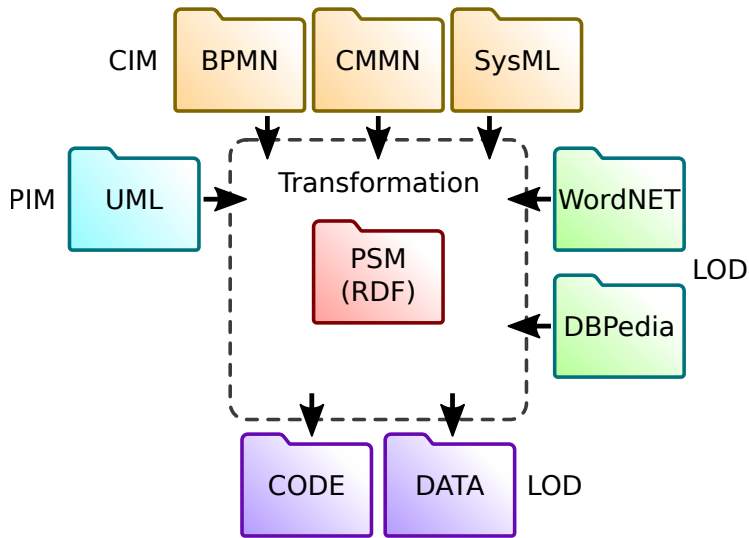
**PM** Platform Model;

**PSM** Platform-Specific Model;

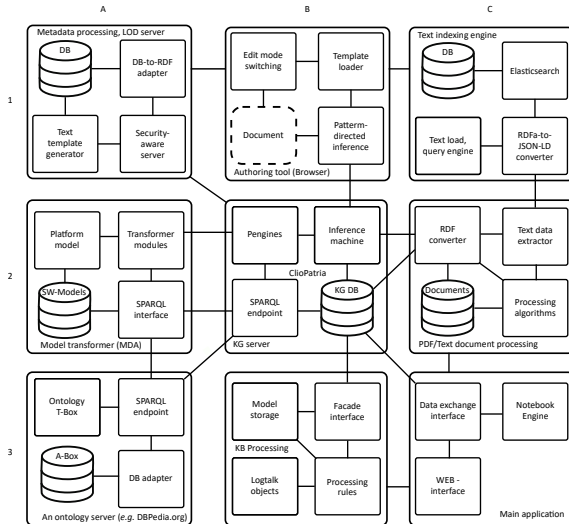
**CODE** Source code of software;

**DATA** Initial database state.

# Model Driven Architecture and Linked Open Data



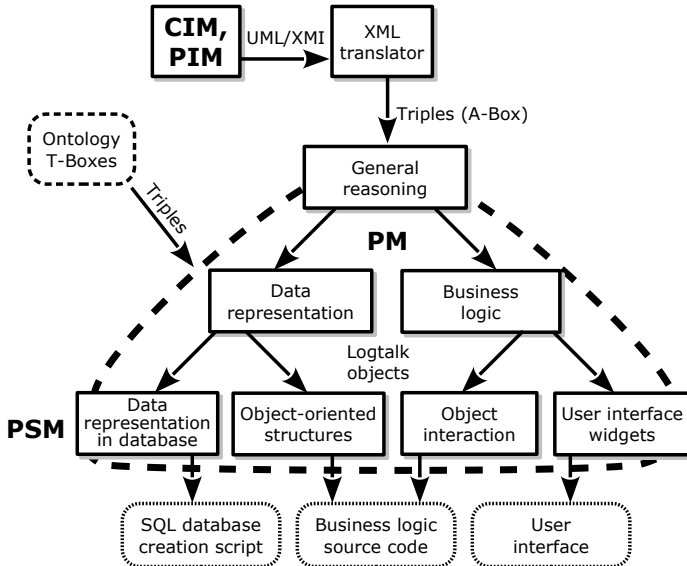
# Architecture of services for NGS



## Abbreviations

T-Module is Transformation module  
MDA is Model-Driven Architecture  
CIM is Computationally Independent Model  
PIM is Platform Independent Model  
PSM is Platform Specific Model  
T-Box is Terminological Box  
A-Box is Instance Box  
NGS is Next-Generation Sequencing  
DB is Database

# Architecture of transformation modules



# Logtalk as transformation definition language

We have chosen Logtalk as it

- ❑ inherits widely known Prolog language syntax and runtime;
- ❑ implemented as macro package, performance penalties are about 1.5%;
- ❑ has flexible semantics: we can define transformations and constraints within the same syntax;
- ❑ implement object-oriented knowledge (rules) structuring, encapsulation and replacement;
- ❑ compositional way of transformation implementation;
- ❑ powerful engine to post constraints on object-to-object messages (events);
- ❑ has implementation for many Prolog engines.

The «regular» language allow us to use its libraries not directly related to MDA transformations.

# RDF (TTL) representation and its query object

```
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
ngsp:spec a ngsp:Specification ;
    ngsp:module mothur:NoCommand,
        mothur:align-check,
        mothur:align-seqs,
# . . . . .
mothur:align-check a ngsp:Module ;
    ngsp:outputPattern [ a cnt:Chars ;
        ngsp:parameterName "type" ;
        ngsp:pattern [ ngsp:patternString
            "[filename],align.check" ;
            dc:identifier "aligncheck" ] ;
        cnt:chars # . . . . .
# . . . . .
mothur:align-check-idir-parameter a ngsp:Parameter ;
    ngsp:important false ;
    ngsp:multipleSelectionAllowed false ;
    ngsp:optionsDefault "" ;
    ngsp:required false ;
    ngsp:type mothur:String ;
    dc:title "inputdir" .

mothur:align-check-map-parameter a ngsp:Parameter ;
    ngsp:important true ;
    ngsp:multipleSelectionAllowed false ;
    ngsp:optionsDefault "" ;
    ngsp:required true ;
    ngsp:type mothur:InputTypes ;
    dc:title "map" .

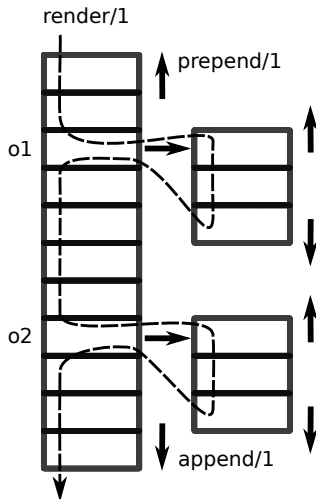
mothur:align-check-name-parameter a ngsp:Parameter ;
    ngsp:chooseOnlyOneGroup "namecount" ;
    ngsp:important false ;
    ngsp:multipleSelectionAllowed false ;
# . . . . .
```

```
:- object(query(_XMI)).
:- protected(xmi/1).
:- public([class/2, attribute/3, method/3]).
xmi(XMI) :- parameter(1, XMI).
    % Recognition of Class in RDF
class(Name, ID):-
    ::xmi(XMI),
    XMI::rdf(ID,rdf:type,uml:'Class'),
    XMI::rdf(ID,rdfs:label, literal(Name)).
    % Recognition of an attribute
attribute(Name, ClassID, ID):-
    ::xmi(XMI),
    XMI::rdf(ClassID, xmi:ownedAttribute, ID),
    XMI::rdf(ID, rdfs:label, literal(Name)).
    % Recognition of a method specification.
method(Name, ClassID, ID):-
    ::XMI(XMI),
    XMI::rdf(ClassID, xmi:ownedOperation, ID),
    XMI::rdf(ID, rdfs:label, literal(Name)).
% . . . . .
:- end_object.
```



# Code Block (idea is taken from llvmlite\*)

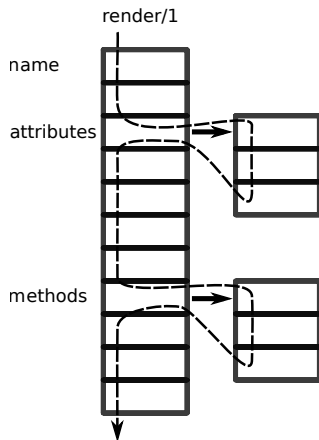
```
:- object(code_block, specializes(root)).
% Public interface of the object
:- public([append/1, prepend/1, clear/0,
          render/1, render_to/1, remove/1,
          item/1, items/1]).
% Code block items
:- dynamic([item_/1]).
:- private([item_/1]).
% Methods specialized during inheritance
:- protected([renderitem/2, render_to/2]).
% . . . . .
% Delegate rendering to object itself
renderitem(Object, String):-
    current_object(Object), !,
    Object::render(String).
% Convert a literal to its string
% representation
renderitem(literal(Item), String):-!,
    atom_string(Item, String).
% Just print the item (debugging).
renderitem(Item, String):-
    root::iswritef(String, '%q', [Item]).
:- end_object.
```



\*) <https://github.com/numba/llvmlite>

# PSM of a Python Class as a specialization of Code Block

```
:- object(class, specializes(code_block),
    imports([named])). % Category of named entities
:- public([classlist/1, methods/1, attributes/1]).
% . . . . .
renderitem(Item, Result):- % proceed with default
    ^^renderitem(Item, Result). % rendering
render(Result):- % Source generator
    ^^render(Name), % implemented in a category
    ( ::item(classlist(List)) ->
        % . . . . .
        [Name]) ),
    ( ::item(attributes(Attributes))->
        % . . . . .
        [DefAttrList]),
    Attributes::items(InstanceAttrs),
    findall(S, ( % initialize attributes
        % . . . . .
        ), AttrAssigns),
    root::unindent,
    AttrList=[ConstructorDef|AttrAssigns];
    % . . . . .
    AttrList=[ConstructorDef, Pass] ),
    ( ::item(methods(Methods))-> % If any ...
        Methods::render(MethodList);
        MethodList=[] ),
    lists::append(AttrList,MethodList,StringList),
    root::unindent, Result=[Signature|StringList].
:- end_object.
```



# Logtalk Categories

## A category of named entities

```
:- category(named).
:- public([name/1, render/1]).
:- protected([renderitem/2]).
name(Name):- ::prepend(name(Name)).
renderitem(name(Name), String):-!, atom_string(Name, String).
render(String):- % What is code generation from items
    ::item(name(Name)), ::renderitem(name(Name), String).
:-end_category.
```

## Category of named and typed entities

```
:- category(namedtyped, extends(named)).
:- public([type/1,render/2, separator_option/2,list_separator/1]).
:- protected([renderitem/2]).
type(Type):- ::append(type(Type)).
renderitem(Item, String):- ^^renderitem(Item, String),!.
renderitem(type(Type),String):-!, ::list_separator(Separator),
    writef::writef(String, '%W%W', [Separator, Type]).
render(Middle, String):- ^^render(SName),
    (    ::item(type(Type)) ->
        ::renderitem(type(Type), SType),
        string_concat(SName, Middle, _1),
        string_concat(_1, SType, String) ;
        SName = String ).
render(String):- ::render("", String).
list_separator(Separator):-
    ::separator_option(Name, Default),!, % Global options
    root::option(Name, Separator, Default).
:- end_category.
```

# Discussion (MDA application)

Interesting positive impressions obtained:

- ❑ Logtalk and RDF are flexible, sufficiently universal and convenient implementation infrastructures for MDA;
- ❑ The best implementation means is Prolog predicate wrapping and Logtalk object encapsulation of rules;
- ❑ Not all Logtalk properties are investigated: there might be more sophisticated programming techniques developed, *e.g.*, on the base of message watchers.

Technical problems making the approach somewhat problematic:

- ❑ Very simple tasks take too much efforts, *e.g.*, text processing: convert an identifier into the CamelCase;
- ❑ It takes too long to surf Internet in order to find a vocabulary for a domain, but it is more productive than development new one and classes;
- ❑ Prolog is not a popular language in MDA, neither Logtalk.

# Application: Document authoring and storage

In most cases documents are created as a result of

- ❑ creative activity of a person with a text processors (authoring);
- ❑ printing a digital copy or a data record in a database;
- ❑ aggregation operation over database records (report).

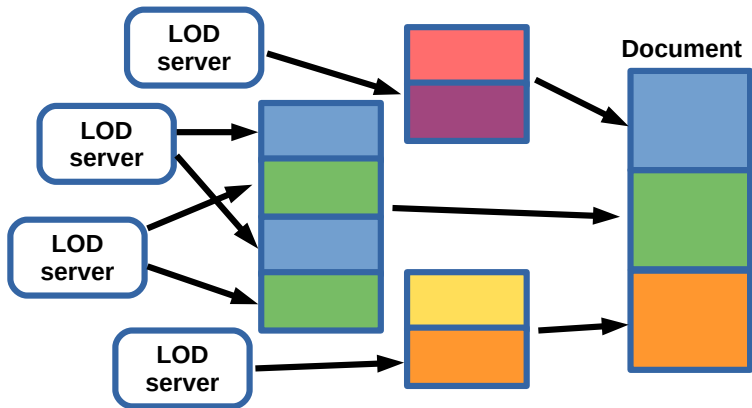
Then it is stored either as a physical paper and/or a digital document (PDF, DOCX, HTML).

Since 2000-th, Semantic Web and Linked Open Data (LOD) is being developed, allowing

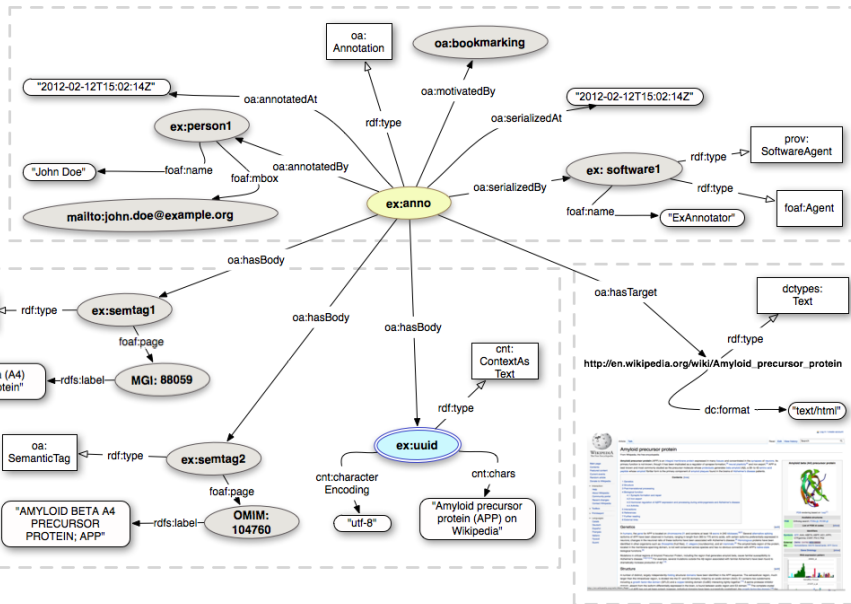
- ❑ structural storage of data within published documents;
- ❑ processing stored data computationally;
- ❑ integration of data structures and data objects globally.

The **aim of this research** is to develop technologies, software and services allowing construction of digital archives supporting document data inclusion and inference from existing documents.

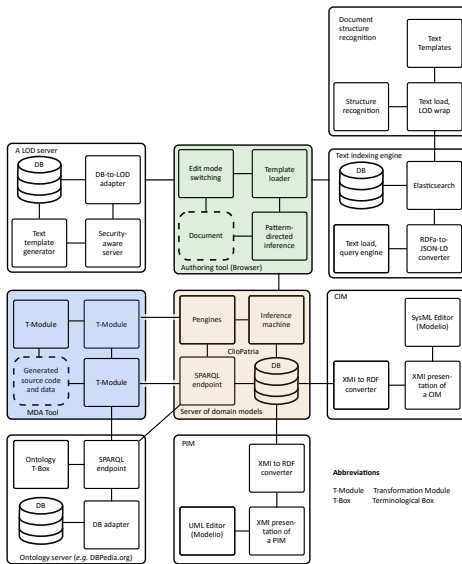
# Structure of a document



# Open Annotation (oa)



# Architecture





# Generated list of title page preambles



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**  
**ФГБОУ ВО «ИГУ»**  
Институт математики экономики и информатики

**Кафедра информационных технологий**



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**  
**ФГБОУ ВО «ИГУ»**  
Институт математики экономики и информатики

**Кафедра алгебраических и информационных систем**

УТВЕРЖДАЮ

## Учебный план специальности 01.03.02 Прикладная математика и информатика

### 1. Общие сведения учебного плана

#### Сведения по Учебному плану

Профиль подготовки: Математическое и компьютерное моделирование в технике и экономике, методы принятия решений

#### Сведения о кафедре, разработавшей Учебный план

Кафедра: Математического анализа и дифференциальных уравнений,  
Факультет: ИМЭИ.

#### Сведения о специальности

Квалификация: Бакалавр

Форма обучения: очная

Программа подготовки: прикладн. бакалавриат

#### Руководители

Проректор по учебной работе: Не распознан

Начальник УМУ: А.И. Вокин

Директор: М.В. Фалалеев

### 2. Список компетенций

#### Дисциплина: Б1.В.ДВ.3.1. Технологии программирования

- способность приобретать новые научные и профессиональные знания, используя современные образовательные и информационные технологии (ОПК-2)
- способность критически переосмысливать накопленный опыт, изменять при необходимости вид и характер своей профессиональной деятельности (ПК-3)
- способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения (ПК-7)

### 3. Список курсов специальности

- Б1.Б.3 «Философия»

# Imported time distribution for lecture, seminary, ...

загрузки,

- методиками экстремального и *agile*-программирования.

## 4. Объем дисциплины (модуля) и виды учебной работы (разделяется по формам обучения)

Вид учебной работы	Всего часов / зачетных единиц	Семестры	
		3	4
Аудиторные занятия (всего)	108	33	75
в том числе:			
Лекции	36		36
Практические занятия (ПЗ)			
Семинары (С)			
Лабораторные работы (ЛР)	66	30	36
КСР	6	3	3
Самостоятельная работа (всего)	45	39	6
в том числе:			

# Representation of document parts with RDFa

```
<html lang="ru" xmlns=http://www.w3.org/1999/xhtml
  xmlns:taa=http://irnok.net/engine/rdfa-manipulation
  xml:lang="ru" metal:define-macro="page">
<head> . . . . </head>
<body prefix="rdf: http://www.w3.org/1999/...-ns# foaf: http://xmlns.com/foaf/...
  imei: imei.html# course: https://irnok.net/college/plan/01..16-...\\
  %D0%BA_PB-SM.plm.xml.xlsx-....2.3.1.html#" resource="#post"
  typeof="schema:CreativeWork sioc:Post prov:Entity">
<!-- The application control panel -->

<main lang="ru" resource="#annotation" typeof="oa:Annotation" id="main-doc-cnt">
<div property="oa:hasTarget" resource="#course-work-prog"></div>
<article property="oa:hasBody" typeof="foaf:Document curr:WorkingProgram"
  resource="#course-work-program" id="main-document">
  <div taa:content="imei:title-page"></div>
  <div taa:content="imei:neg-UMK"></div>
  <section id="TOC" class="break-after"> <h2>Table of Contents</h2>
    <div id="tableOfContents"></div>
  </section>
  <section id="course-description" resource="#description"
    property="schema:hasPart" typeof="schema:CreativeWork">
    <div property="schema:hasPart" resource="#purpose"
      typeof="dc:Text cnt:ContentAsText" >
      <div property="cnt:chars" datatype="xsd:string">
        <h2 property="dc:title" datatype="xsd:string">
          Aims and objectives of the discipline (module)</h2>
        <p>The aim of teaching the discipline ...</p>
      </div>
    </div>
  . . . . .
```

# Complete document



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ**  
**федеральное государственное бюджетное образовательное учреждение  
высшего образования**  
**«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**  
**ФГБОУ ВО «ИГУ»**  
**Институт математики экономики и информатики**

**Кафедра информационных технологий**

УТВЕРЖДАЮ

Директор ИМЭИ

" " 20 г.

**Рабочая программа дисциплины (модуля)**  
**Б1.В.ДВ.3.1. Технологии программирования**

Направление подготовки:	10.03.01 (090900) Информационная безопасность
Направленность (профиль)	- общий
Квалификация (степень) выпускника	- бакалавр
Форма обучения	- очная

Иркутск 2016 г.

Согласовано с УМК факультета (института)

Рекомендовано кафедрой:

Протокол № " " 20 г.

Протокол № " " 20 г.

Председатель  
(подпись)

Зав. кафедрой  
(Ф.И.О.)

**Содержание**

1. Цели и задачи дисциплины (модуля)
2. Место дисциплины в структуре ОПОП
3. Требования к результатам освоения дисциплины (модуля)
4. Объем дисциплины (модуля) и виды учебной работы (разделяется по формам обучения)
5. Содержание дисциплины (модуля)
6. Перечень семинарских, практических занятий и лабораторных работ
7. Примерная тематика курсовых работ (проектов)
8. Учебно-методическое и информационное обеспечение дисциплины (модуля)
9. Материально-техническое обеспечение дисциплины (модуля)
10. Образовательные технологии
11. Оценочные средства (ОС)

**1. Цели и задачи дисциплины (модуля)**

Цель преподавания дисциплины «Технологии программирования» является освоение студентами практических навыков в области разработки программного обеспечения на основе современных подходов к проектированию сложных, гетерогенных, распределенных информационных систем. Развитие навыков системного мышления, необходимого для

# Discussion

A tools (components) for digital archive implementation, which allows to device information systems and document processing services with the following features:

- ❑ load LOD marked up document, extract, store in a graph and index RDF data;
- ❑ retrieve RDF data as triples or as a result of full-text search query;
- ❑ combine existing LOD data and its content in new documents dynamically with browser based context inference machine;
- ❑ use server-site inference machine (Prolog) to process RDF data upon request from browser's part of the system;
- ❑ convert created RDFa marked up HTML5 documents into Excel and Word formats.

## Applications

- ❑ Document authoring automation;
- ❑ Context-depended editing;
- ❑ Self-organizing global document flows;
- ❑ Documents as data sources for information systems.

# Application: Cartographical WEB-service with knowledge graph of South-Siberian faults

**Aim** is to construct a WEB-GIS browser for faults stored in the KG.

- ❑ Scalability to external data with converters (TODO)
- ❑ Interdisciplinary data representation
- ❑ Application development with nowadays WEB techniques
- ❑ Digital platform for data publication in “Digital Baikal” project

**Development plan**

- ❑ Investigate the current data formats
- ❑ Develop T-Box
- ❑ Fill in A-Box
- ❑ Expose the KG with a server
- ❑ Implement browsing SPARQL query results with GIS
- ❑ Develop object browser

MVP is a WEB-GIS with the most of the listed features.

# GIS source data table properties

- Only one table, one row for each GIS object
- There are many NULL values
- More than 1000 objects
- More than 70 attributes (according to O.V. Lunina, PhD)

ActiveFault\_reliable - Всего объектов: 1163, Отфильтровано: 1163, Выделено: 0

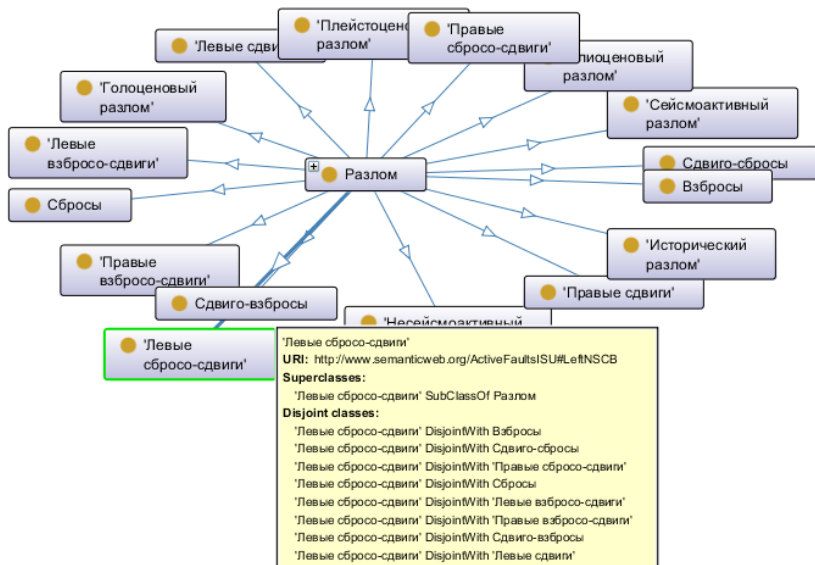
idno	Total_Ce_1	Total_Ce_2	Total_Ce_3	Reliability	Geomorphol	Geomorph_1	Geophysica	Geophys_1	Engineerin	Engineer_1	Gydrogeolo	Gydrogeo_1	Meteorolog	Meteorol_1	Structur
1	ЛС	NULL	ЛС	1	Серия линейн...	3 NULL			0 При Културско...	0	0 После землетр...	2 NULL	0	0 Вбросово...	
2	ЛС	300	ЛС	1	Линеймент на ...	3 NULL			0 NULL		0 NULL	0 NULL	0 NULL	0 NULL	
3	ЛС	450	ЛС	1	Линеймент на ...	3 NULL			0 Сейсмограви...		3 NULL	0 NULL	0 NULL	0 NULL	
4	ЛС	700	ЛС	1	Линеймент на ...	3 NULL			0 NULL		0 NULL	0 NULL	0 NULL	0 NULL	
5	ЛС	2000	ЛС	1	Уступ на топо...	3 Сетчатое разло...			1 NULL		0 NULL		0 NULL	0 Многочисл...	
6	ЛС	2000	ЛС	1	Уступ на топо...	3 Выделяется в г...			1 Обвалы объем...		3 Восходящий в...	2 NULL	0 Крутой скл...		
7	ЛС	2000	ЛС	1	Уступ на топо...	3 NULL			0 NULL		0 NULL	0 NULL	0 NULL	0 NULL	
8	ЛС	2000	ЛС	1	Уступ на топо...	3 NULL			0 NULL		0 NULL	0 NULL	0 NULL	0 NULL	
9	ЛС	2000	ЛС	1	Уступ на топо...	3 NULL			0 NULL		0 NULL	0 NULL	0 NULL	0 NULL	
10	ЛС	2000	ЛС	1	Уступ на топо...	3 NULL			0 NULL		0 NULL	0 NULL	0 NULL	0 NULL	
11	NULL	NULL	NULL	1	Уступ и линее...	3 NULL			0 NULL		0 NULL	0 NULL	0 NULL	0 Зоны трещи...	
12	NULL	NULL	NULL	1	Линеймент на ...	1 NULL			0 NULL		0 NULL	0 NULL	0 NULL	0 NULL	
13	NULL	NULL	NULL	1	Линеймент на ...	1 NULL			0 NULL		0 NULL	0 NULL	0 NULL	0 NULL	
14	NULL	NULL	NULL	1	Линеймент на ...	3 NULL			0 Крутые сейс...		3 NULL	0 NULL	0 NULL	0 NULL	
15	NULL	NULL	NULL	1	Линеймент на ...	3 Изгибы кровл...			1 NULL		0 Восходящий и...	2 NULL	0 Сейсмиче...		
16	NULL	NULL	NULL	1	линеймент на ...	3 NULL			0 Обвалы объем...		3 NULL	0 NULL	0 NULL	0 NULL	
17	NULL	NULL	NULL	1	Серия линейн...	1 NULL			0 NULL		0 NULL	0 NULL	0 NULL	0 NULL	
18	NULL	25	ЛС	1	Линеймент на ...	3 NULL			0 Крутые сейс...		3 NULL	0 NULL	0 NULL	0 NULL	
19	NULL	NULL	NULL	1	Линеймент на ...	1 NULL			0 NULL		0 NULL	0 NULL	0 NULL	0 NULL	
20	NULL	NULL	NULL	1	Серия линейн...	1 NULL			0 NULL		0 Горючий источ...	2 NULL	0 NULL	0 NULL	
21	NULL	NULL	NULL	1	Изгибы рек на ...	1 NULL			0 NULL		0 NULL	0 NULL	0 NULL	0 NULL	
22	NULL	NULL	NULL	1	Серия линейн...	1 NULL			0 NULL		0 NULL	0 NULL	0 NULL	0 NULL	
23	NULL	NULL	NULL	1	Линеймент на ...	1 NULL			0 NULL		0 NULL	0 NULL	0 NULL	0 NULL	
24	NULL	NULL	NULL	1	Отдельные сла...	1 NULL			0 NULL		0 NULL	0 NULL	0 NULL	0 NULL	
25	NULL	NULL	NULL	1	Серия выветр...	1 NULL			0 NULL		0 NULL	0 NULL	0 NULL	0 NULL	

1 Все объекты



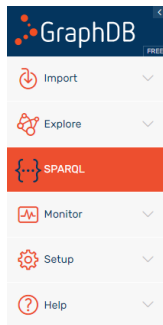
# Developed ontology

The ontology contains nonintersection properties for its classes



# Serving ontology and its A-box

As server GraphDB is used.



GraphDB FREE

- Import
- Explore
- {...} SPARQL**
- Monitor
- Setup
- Help

## SPARQL Query & Update

testFault

```
1 PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
2 PREFIX : <http://192.168.2.2:7200/repositories/ActiveFaults#>
3 SELECT ?s ?id
4 WHERE
5 {
6     ?s rdf:type :Fault.
7     ?s rdf:type :HoloceneFault.
8     ?s rdf:type :ShiftUpCB.
9     ?s :Name ?id
10 }
```



Run

Press Alt+Enter to autocomplete

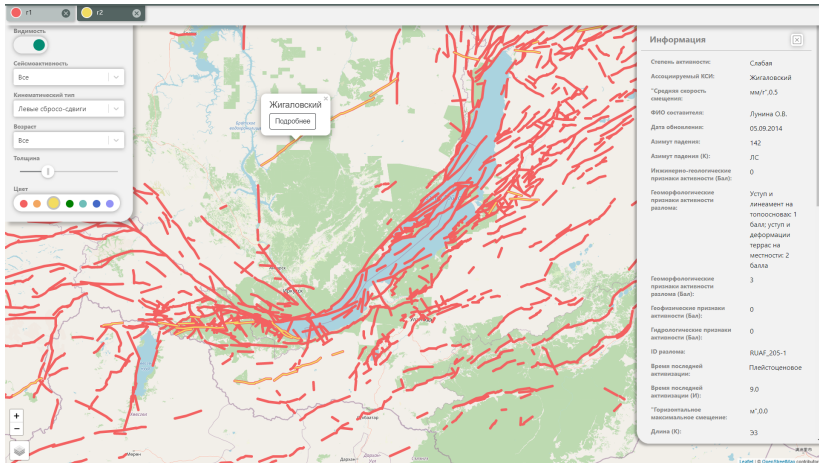
Table Raw Response Pivot Table Google Chart

Download as

Filter query results

Showing results from 1 to 5 of 5. Query took 0.1s, moments ago.

	s	id
1	:MNAF__1746	"Хиргиснурский-1"
2	:RUAF__1668-6	"Каахемский"
3	:RUAF__1668-7	"Каахемский"
4	:RUAF__1718	"Джебашский"



# Used technologies for constructing WEB-GIS browser



# Ontological instruments: editor Protégé

Active Ontology x Entities x Classes x Object Properties x D

Class hierarchy (Inferred)  
Class hierarchy

Class hierarchy: Feature

Annotations

Asserted

- owl:Thing
  - Fault
    - Associated\_CSS
    - Associated\_IGGSS
    - Info
      - Compiler
      - Date
      - Location
      - Name
      - Reliability\_class
    - Last\_historical\_earthquake
    - Seismic\_activity\_of\_fault
  - Quality
    - ActivityDegree
    - EventParameter
    - FeatureQuality
    - Unit
  - Value
    - Angular
    - Event
    - Linear
      - Depth
      - Length
      - Rate
      - Width
    - Notional
      - Feature
      - Slip
        - Cenozoic\_Slip

Datatypes

Annotation property hierarchy

Individuals by type

Data property hierarchy

Object property hierarchy

Object property hierarchy: feature

Asserted

- owl:topObjectProperty
  - property
    - cenozoic
    - qualityProperty
    - quantityProperty
    - value
      - activity
        - feature

```
[eugeneai@center data]$ head -n 50 activity_fall_data.ttl
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix geob: <http://www.semanticweb.org/bernard_black/ontologies/2016/3/
@prefix nie: <http://www.semanticdesktop.org/ontologies/2007/01/19/nie#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

[] a geob:Fault ;
  nie:identifier "RUAF_235-1" ;
  nie:title "Северобайкальский" ;
  geob:activity [ a geob:Activity ;
    geob:degree "повышенная" ;
    geob:eventage [ a geob:EventAge ;
      geob:index 2e+00 ;
      geob:type geob:LastActivationAge ;
      geob:value "Голоценовое" ] ;
      geob:value 1.4e+01 ] ;
  geob:angle [ a geob:Angle ;
    geob:quality "ЛС" ;
    geob:value "50-80" ] ;
  geob:azimuth [ a geob:Azimuth ;
    geob:quality "ЛС" ;
    geob:value 1.02e+02 ] ;
  geob:cenozoicSlip [ a geob:CenozoicSlip ;
    geob:quality "ЛС" ;
    geob:reliabilityClass 1e+00 ;
    geob:type geob:vertical ;
    geob:value 4e+03 ] ;
  geob:compiler [ a geob:Compiler ;
    nie:created "15.11.2014" ;
    foaf:name "Лунина О.В." ] ;
  geob:event [ a geob:Event ;
    geob:associatedCSS "Северобайкальский" ;
    geob:averaged_slip_rate_mm_year 3e+00 ;
    geob:isActiveFault "Да" ;
    geob:potential_ms_max 7.7e+00 ;
    geob:potential_mm_max 0e+00 ;
    geob:quality "ЛС" ;
    geob:slip_rate_mm_year "1-4.99" ;
    geob:type geob:TotalMaxSlip ;
    geob:value 9e+00 ],
  [ a geob:Event ;
    geob:type geob:LateralMaxSlip ;
    geob:value 0e+00 ],
  [ a geob:Event ;
    geob:type geob:VerticalMaxSlip ;
    geob:value 9e+00 ] ;
  geob:feature [ a geob:Feature ;
    [eugeneai@center data]$
```

# Modification of GeoBase supporting Semantic WEB

```
schema('fault','in','continent'). % Connect our relations with GeoBase
schema('fault','with','feature'). % vocabulary.
schema('name','of','fault').      %
% schema('feature','of','fault'). % This relation is already in the T-Box''
```

*% Interpret any well described relation*

*% between a subject (fault) via 'of'.*

```
schema(Prop, 'of', SubjName):-      % used on translation stage
    var(SubjName),
    geob_prop(Prop,_).
```

```
schema(Prop, 'of', SubjName):-      % used on stage of interpretation
    nonvar(SubjName),               % a Class is supplied
    geob_prop(Prop, GProp),
    geob_ent_class(SubjName, Subj),
    rdf_reachable(Subj, rdfs:subClassOf, Parent),
    rdf(GProp, rdfs:domain, Parent),!.
```

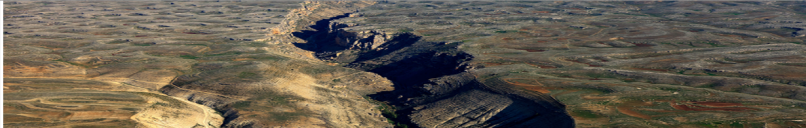
```
geob_prop(Prop, GProp):-            % Property check
    rdf_global_id(geob:Prop, GProp),
    rdf(GProp,rdf:type,owl:'ObjectProperty'),!.
```

```
geob_class(Class, GClass):-         % Class check
    rdf_global_id(geob:Class, GClass),
    rdf(GClass,rdf:type,owl:'Class'),!.
```

# GeoBase to ActiveFaults Natural language interface

ubuntu:3020/apps/geobase/index.html

Search



Введите запрос

show any names of faults

Go Clear

Результат

Entity	Type
Аблатуканский	name
Абчадский	name
Агайский	name
Аганайский	name
Агардагский	name
Агардагский (Эрзино-Агардагский)	name
Агардагский (Эрзинско-Агардагский)	name

**New problem** for student graduation project: Implement Natural Language to SPARQL translator.

# Conclusion

Web-3.0 (SW and KG) are convenient and productive basis of application development, allowing one to

- ❑ Integrate on various levels of software representation, *i.e.*, from data to abstract models.
- ❑ There are assets to utilize global resources in applications (DBPedia.org).
- ❑ The inherited software are involved via adapters.

In our projects we has been developed techniques to

- ❑ Data processing and model transformation using Prolog-based logical inference.
- ❑ Logtalk is a perfect instrument for knowledge encapsulation and processing, complex data synthesis.



Thanks for Your Attention!