

Construction techniques of Baikal microbiome research information-computational environment

Evgeny Cherkashin, Alexey Shigarov, Vasily Khristyuk

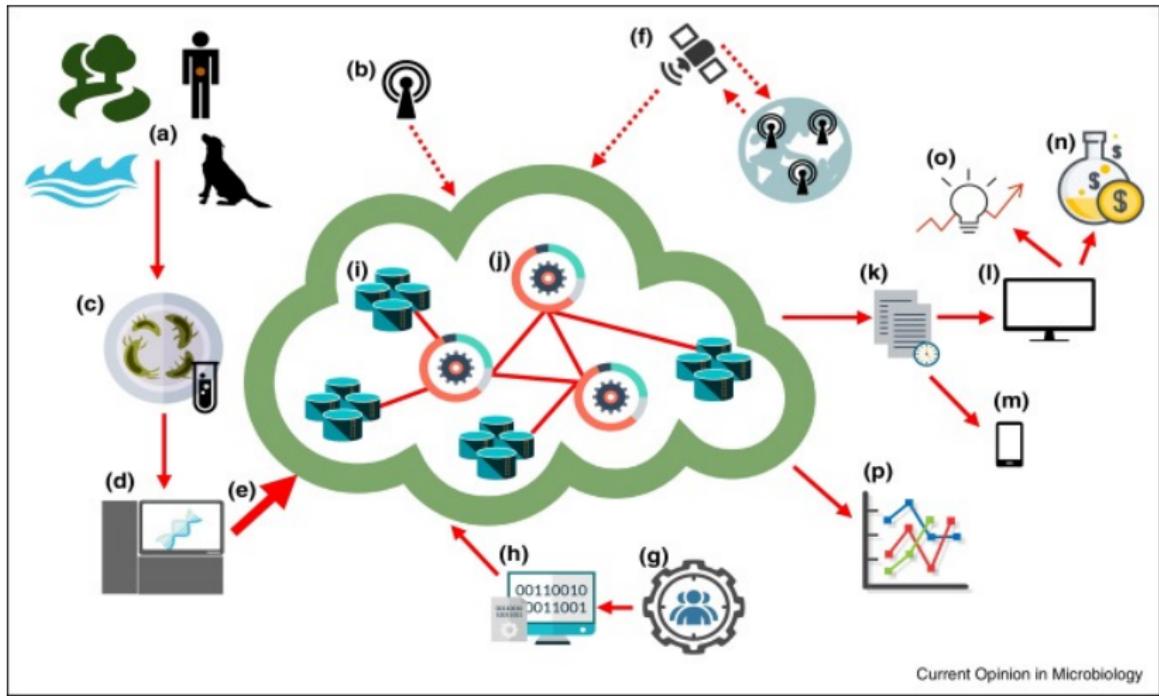
Matrosov Institute for System Dynamics and Control Theory,
SB RAS, Irkutsk, Russia

Irkutsk Scientific Center SB RAS, Irkutsk, Russia
Irkutsk State Technical University, Irkutsk, Russia

eugeneai@icc.ru

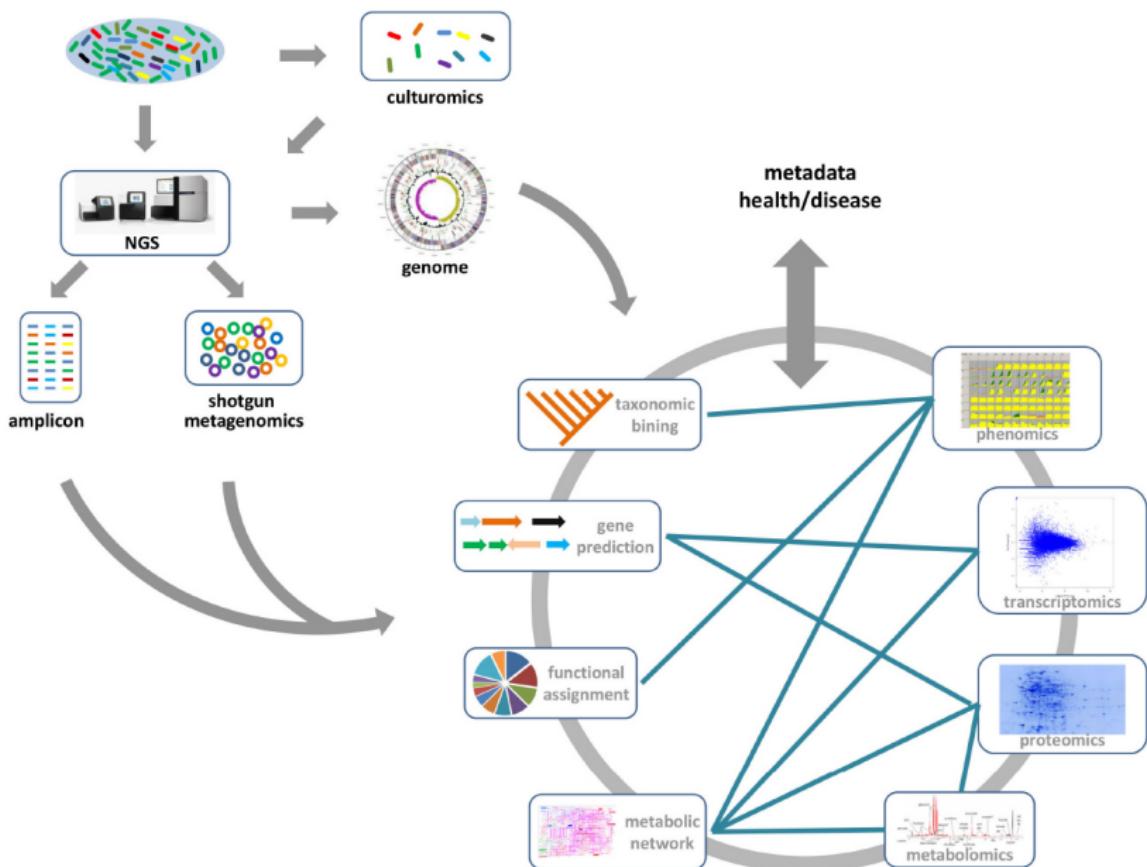
AICTS-2020, Irkutsk, Russia

Microbiome



Current Opinion in Microbiology

Microbiome study



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 SOP¹ (a technique) and comprehensive representation.

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;
- ❑ result and intermediate data visual representation;
- ❑ data integration unit based on Semantic Web and Linked Open Data principles.

¹Standard Operational Procedure

The process of *amplicon* data analysis (MiSeq SOP)

1. Reconstruct *cotigs* (contiguous gene parts) from “left” and “right” *readings*.
2. Trim **bar-code** and other *primers*.
3. Filter sequences according to formal criteria (ambiguity, average length, maximal length of homopolymer).
4. Classify *unique* sequences and count their appearance in groups (samples).
5. Alignment with reference sequences from SILVA database.
6. Filter non-hanging sequences.
7. Filter chimeras, fund unique sequences again.
8. Classify sequences with respect to existing taxa hierarchy. Get **OTUs**.

After these stages around 1000 OTU² has been obtained.

²Operation Taxonomic Unit

Alignment (an example)

```
align.seqs(fasta=HXH779K01.shhh.trim.good.unique.fasta,  
reference=silva.bacteria.fasta)
```

```
HXH779K01.shhh.trim.good.unique.align  
HXH779K01.shhh.trim.good.unique.align.report  
HXH779K01.shhh.trim.good.unique.flip.accnos
```

```
summary.seqs(fasta=HXH779K01.shhh.trim.good.unique.align,  
count=HXH779K01.shhh.trim.good.count_table)
```

	Start	End	NBases	Ambigs	Polymer	NumSeqs
Minimum:	6237	22529	5	0	1	1
2.5%-tile:	6428	22577	409	0	4	835
25%-tile:	6428	25292	420	0	4	8349
Median:	6428	25292	423	0	5	16697
75%-tile:	6428	25292	442	0	5	25045
97.5%-tile:	6430	25293	446	0	6	32558
Maximum:	43107	43116	476	0	9	33392
Mean:	6435	25061	429	0	4	

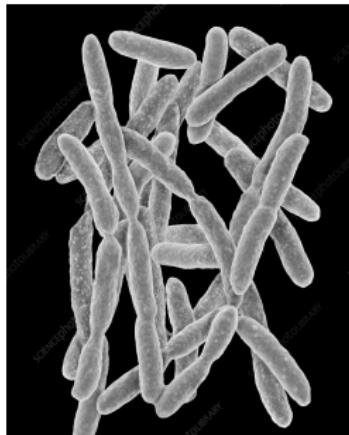
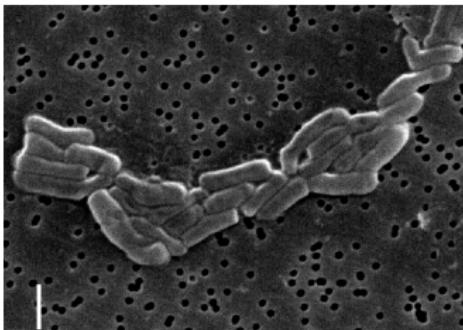
```
# of unique seqs: 26744  
total # of seqs: 33392
```

It took 21 secs to summarize 33392 sequences.

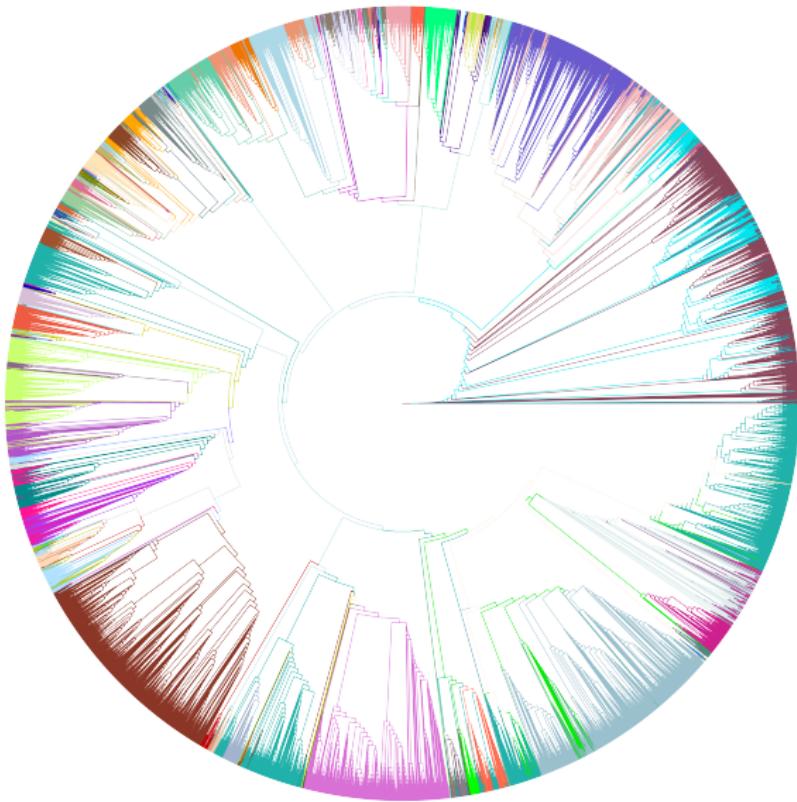
OTU classification analysis (an example)

```
classify.otu(list=HXH779K01.shhh.trim.good.unique.good.filter.unique.precluster.pick.pick.opti_mcc.list,
count=HXH779K01.shhh.trim.good.unique.good.filter.unique.precluster.denovo.vsearch.pick.pick.count_table,
taxonony=HXH779K01.shhh.trim.good.unique.good.filter.unique.precluster.pick.rdp.wang.pick.taxonomy,
label=0.03)
```

OTU	Size	Taxonomy
Otu001	2883	Bacteria(100);Bacteria_unclassified(100);Bacteria_unclassified(100); Bacteria_unclassified(100);Bacteria_unclassified(100);Bacteria_unclassified(100);
Otu003	1038	Bacteria(100);"Actinobacteria"(100);Actinobacteria(100);Actinomycetales(100); Actinomycetales_unclassified(100);Actinomycetales_unclassified(100);
Otu005	660	Bacteria(100);"Actinobacteria"(100);Actinobacteria(100);Acidimicrobiales(100); Acidimicrobiaceae(99);Illumatobacter(99);
Otu007	497	Bacteria(100);"Verrucomicrobia"(100);Subdivision3(100);Subdivision3_..._sedis(100); Subdivision3_..._unclassified(100);Subdivision3_genera_incertae_sedis_unclassified(100);
Otu008	484	Bacteria(100);"Bacteroidetes"(100);Flavobacteria(100);"Flavobacteriales"(100); Flavobacteriaceae(100);Flavobacterium(100);

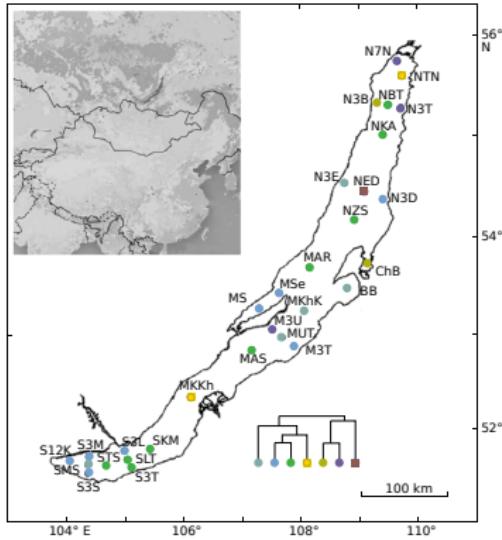
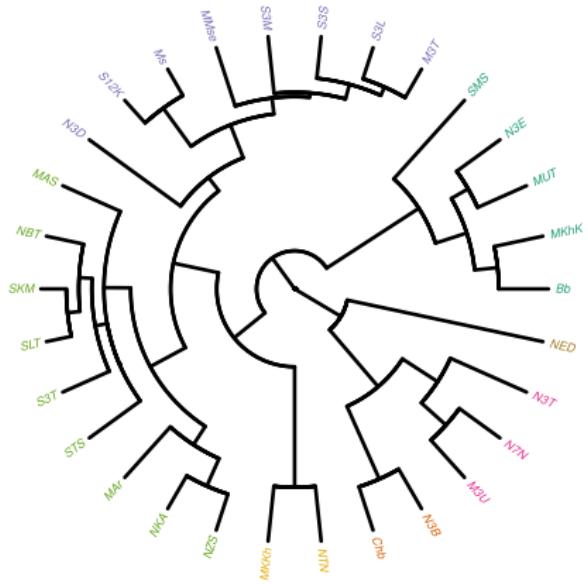


Phylogenetic OTU investigations

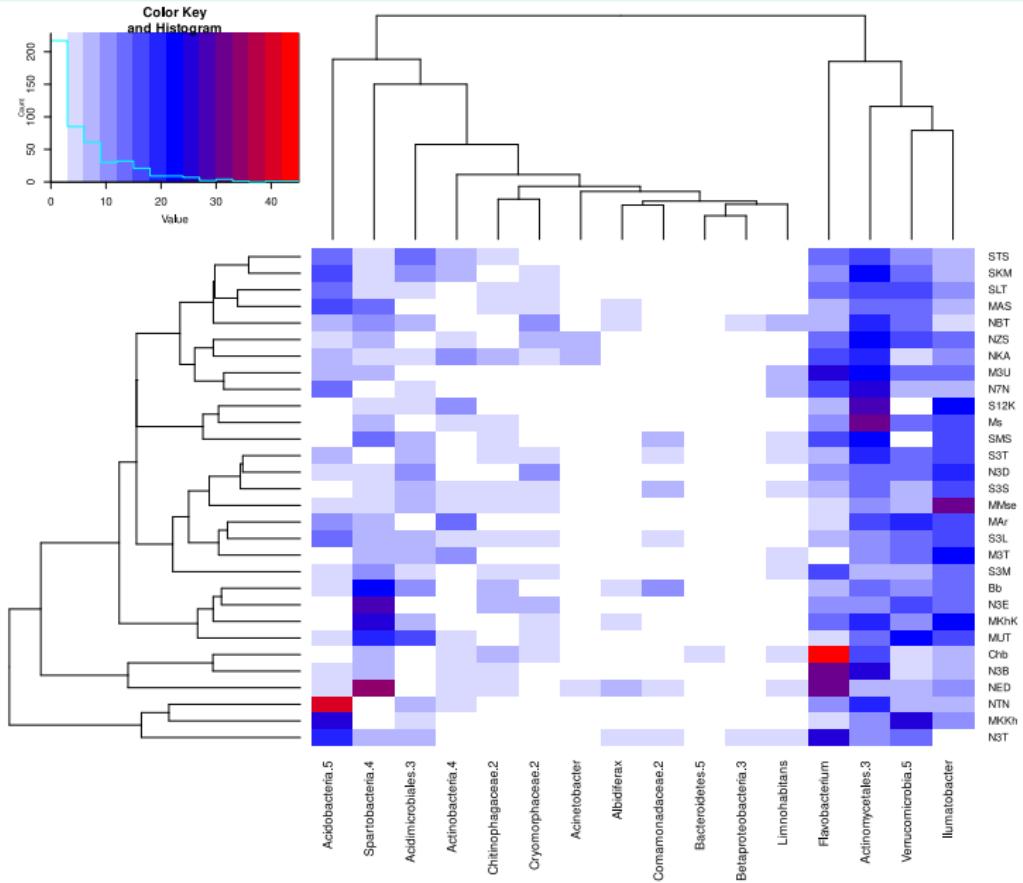


A “taxonomy” of groups

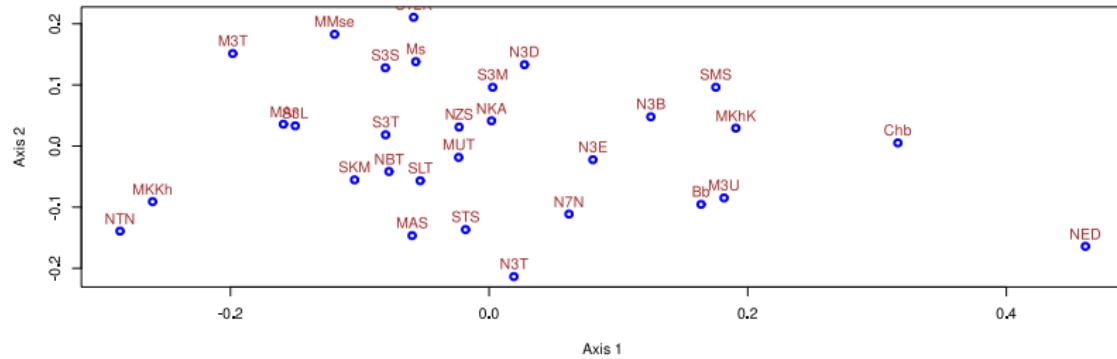
The taxonomy is build using unique sequences as a comparison basis.



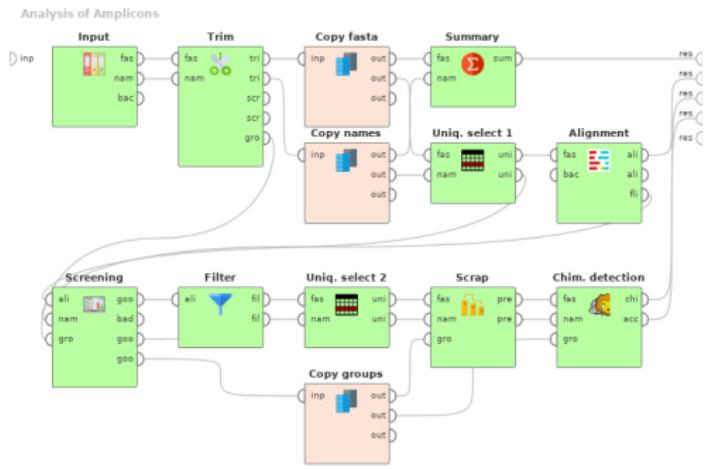
Alpha-diversity



Beta-diversity



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 Studio
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);
    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(pmacth);
    // . . . .
}

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" .
# . . .
```

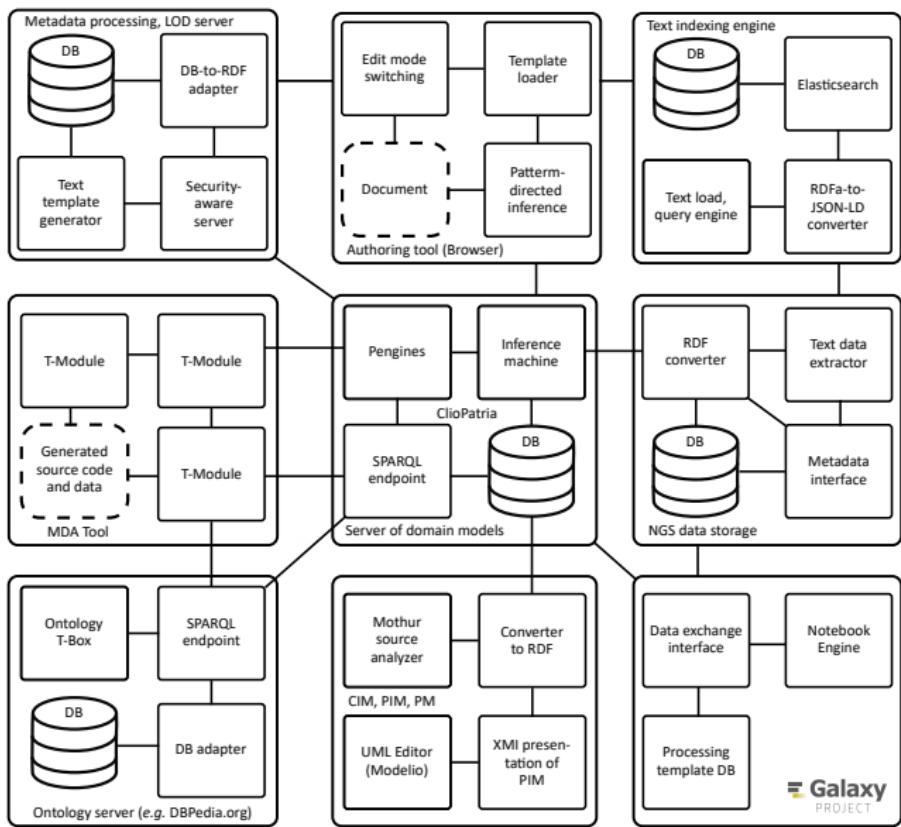
Procedural data (Mothur tooling of Galaxy)

```
<tool profile="16.07" id="mothur_make_contigs"
  name="Make.contigs" version="@WRAPPER_VERSION@"
  <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}'
)' | sed 's/ //g' | mothur | tee mothur.out.log
]]></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 to all" ;
          dc:title "pdiffs" ;
          rdfs:range "integer" ],
          [ gal:min "0" ;
            gal:value "0" ;
            dc:description "bdiffs - number of differences to a" ;
            dc:title "bdiffs" ;
            rdfs:range "integer" ],
          [ gal:min "0" ;
            gal:value "0" ;
            dc:description "tdiffs - total number of difference" ;
            dc:title "tdiffs" ;
            rdfs:range "integer" ] ] ] ] ;
      dc:identifier "mothur_make_contigs" ;
      dc:title "Make.contigs",
        "make.contigs" ;
      schema:sku 1 ''
```

The infrastructure



Semantic Web technologies in representation of models during transformation

- ❑ Assimilates experience of domain basic researches trending to standardization;
- ❑ Regular set of triples denote a graph (T-Box, A-Box);
- ❑ Standard vocabularies are formally described (`rdfs:domain`, `rdfs:range`);
- ❑ Supported with most programming systems (libraries, inference engines, SPARQL);
- ❑ RDF has a way of global element identification, *i.e.* we can refer the same object from different software systems;
- ❑ SWI-Prolog supports direct queries to a graph, as well as interpreting some predicates (`rdfs:label`, `dc:title`), wraps sparse RDF structure into a predicate arguments; ontological server ClioPatria;
- ❑ There is simple way of data security implementation (`rdfs:seeAlso`);
- ❑ By means of Semantic Web & LOD we are able to organize data transfer between heterogeneous information systems.

Used ontologies

Standardized ontologies

- ❑ 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.
- ❑ Schema.org (**schema**) for Google, Yandex, Yahoo, *etc.* searchable objects, structural elements.
- ❑ The Bibliographic Ontology (**bibo**) used for literature reference mark up.
- ❑ Open annotation (**oa**) as an “bookmark” ontology.

Non-standard ontologies

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

PSM: Scenario of a Class synthesis

```
:- object(direct(_Package,_LocalProf,_CodeProf)).  
:- public([tr/4,tr/3]).  
% . . . . .  
tr(class, Class, ClassID):- ::package(Package),  
    query(Package)::class(Name, ClassID),  
    create_object(Class, % . . . . .  
    create_object(Attributes, % . . . . .  
    create_object(Methods, % . . . . .  
Class::name(Name),  
    % Generate attributes of the class,  
    % organizing them in a local database.  
    % ...methods...  
    Class::attributes(Attributes),  
    Class::methods(Methods).  
  
tr(attribute, Attribute, ClassID, AttributeID):-  
    ::package(Package),  
    query(Package)::attribute(Name,ClassID,AttrID),  
    create_object(Attribute, % . . . . .  
Attribute::name(Name).  
  
tr(method, Method, ClassID, MethodID):-  
    ::package(Package),  
    query(Package)::method(Name,ClassID,MethodID),  
    create_object(Method, % . . . . .  
Method::name(Name)).  
:- end_object.
```

% Transformation driver object
% Public interface of a class synthesis scenario

% Synthesize a class
% Query package structure in XMI
% Create a «Class» object
% Create «Attributes» obj
% ...«Methods».
% Name the class.

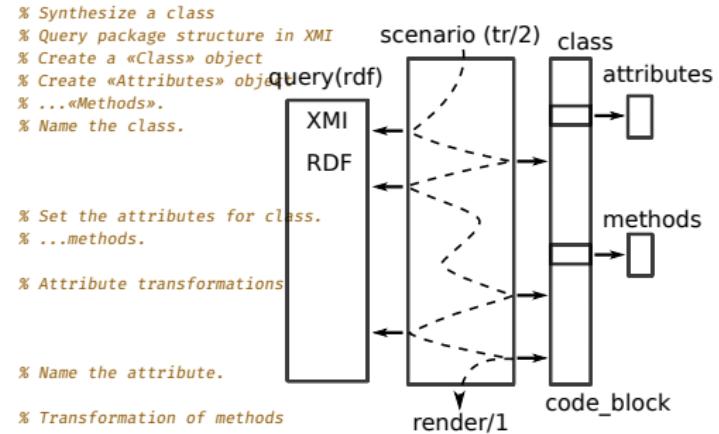
% Set the attributes for class.
% ...methods.

% Attribute transformations

% Name the attribute.

% Transformation of methods

% Name of the method

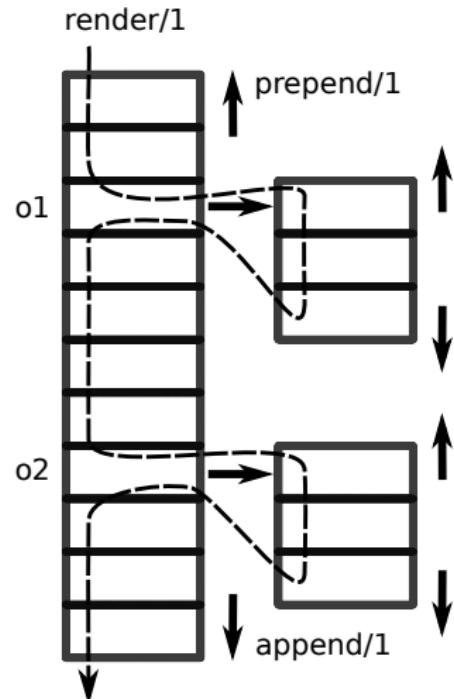


Implementation of Query object

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

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.

Future activities

The future activities supposed to be as follows:

1. Having dataflow models of MiSeq SOP and other techniques, device an intelligent subsystem, which will construct computational procedures for a predefined set of data processing tasks (*AI's problem solving*).
2. Integrate dataflow and data storage with Galaxy.
 - ▶ Realize adapters of data/metadata storage and retrieving, as well as the storage.
 - ▶ Create a more sophisticated source code parser or PIM model of computation so we will able to infer metadata for the output on the metadata of input (partially done in November).
 - ▶ Adapt our document authoring tools to Galaxy allowing LOD representation of results.
3. Implement integration to biological/gene databases.
4. Write a handbook on Logtalk programming strategies with its author Paulo Mora.

Conclusion

The following results have been obtained as for today:

- ❑ Biologists' activities are mastered, investigated and regular patterns are described.
- ❑ A technique for interpretation of Mothur interfaces has been developed, implemented, refined and extended.
- ❑ Transformation tools are tested in application areas and no significant technical problems were detected.
- ❑ A technique of document authoring is being developed and adapted to the domain.
- ❑ Integration with Galaxy project is the primary aim of the future development.

The source codes are available at

<https://github.com/isu-enterprise/icc.xmittransform>,
<https://github.com/eugeneai/icc.mothurpim>.

This research is supported by Irkutsk scientific center of SB RAS, project No 4.2;

Technologies used (open source)

Python-3.x.x (<http://python.org>),
ZCA (<https://muthukadan.net/docs/zca.html>),
SWIG (<http://swig.org/>),
SWI-Prolog (<https://www.swi-prolog.org/>),
Logtalk (<https://logtalk.org/>),
ClioPatria (<https://cliopatria.swi-prolog.org/home>),
Pengines (<https://pengines.swi-prolog.org/docs/index.html>),
LOV (<https://lov.linkeddata.es/dataset/lov/>),
Elastic Search (<https://www.elastic.co/>),
Kyotocabinet (<https://fallabs.com/kyotocabinet/>),
DBPedia (<https://wiki.dbpedia.org/>),
Mothur (<https://mothur.org/>),
Galaxy (<https://usegalaxy.org/>),
R (<https://www.r-project.org/>),
Dust.js (<https://akdubya.github.io/dustjs/>)

External links



This presentation

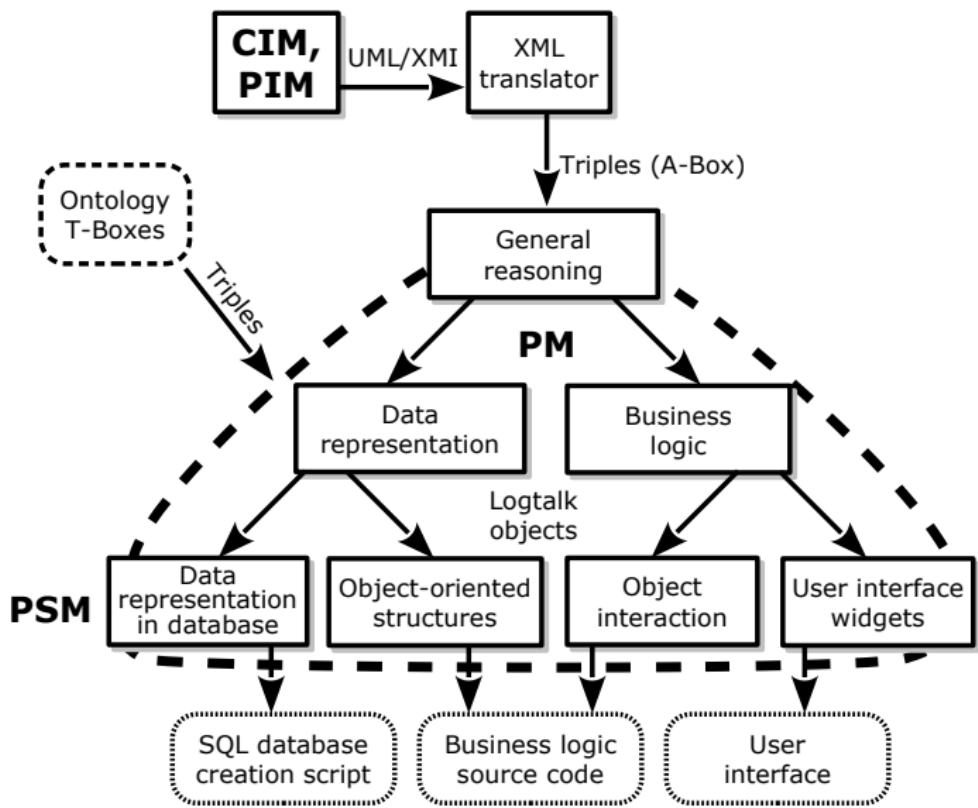


Illumina movie on NGS

Thanks for Your interest in our project!

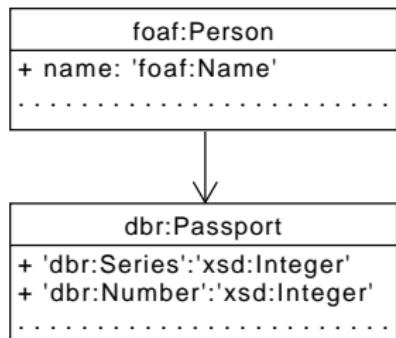
Auxiliary materials

Architecture of transformation modules



Access LOD data

```
:- category(sparql).
:- public(query/2).
query(Pattern,Parameters,Row):-  
    prepare(Pattern,Parameters,Query),  
    server(Host,Port,Path),  
    sparql_query(Query, Row,  
        [host(Host),port(Port),path(Path)]).  
:- protected(server/3). % must be implemented  
                         % by a subclass.  
:- protected(prepare/3). % prepares a query  
% . . . . . . . . . . . % string.  
:- end_category.  
  
:- object(dbpedia, extends(sparql)).  
:- protected(server/3).  
server('dbpedia.org',80,'/sparql').  
:- public(entity_name/2).  
entity_name(Entity,Language,Name):=  
    query('select ?name where {  
        ?w rdfs:label ?name.  
        FILTER langMatches( lang(?label),  
            "%w" )}', [Entity, Language],  
        row(Name)).  
:- end_object.  
  
% ?- dbpedia::entity_name(dbr:'Passport', 'ru', Name).
```



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).

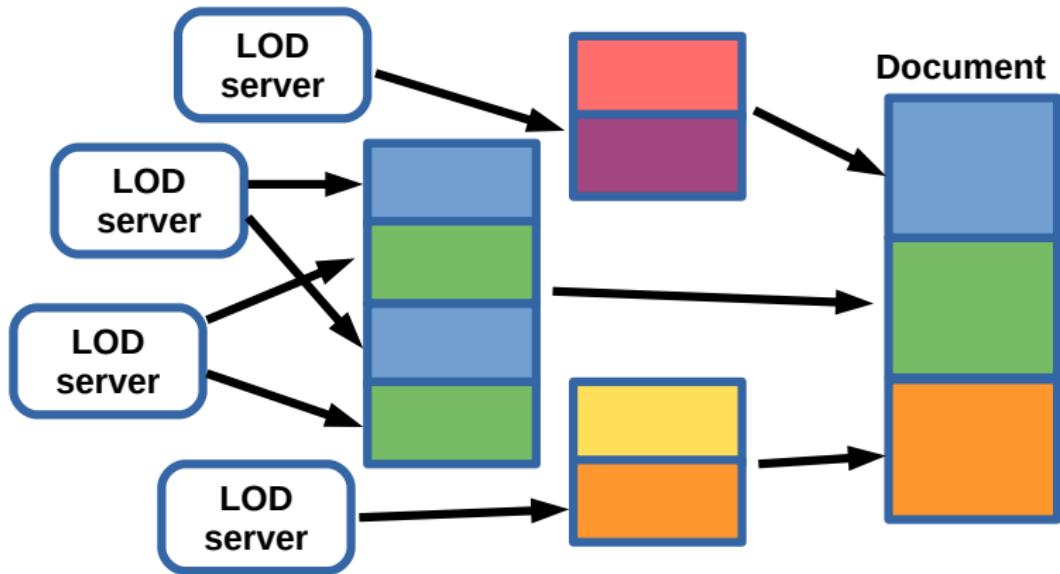
Then it is stored either as a physical paper and/or a digital document (PDF, DOCX, HTML).

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

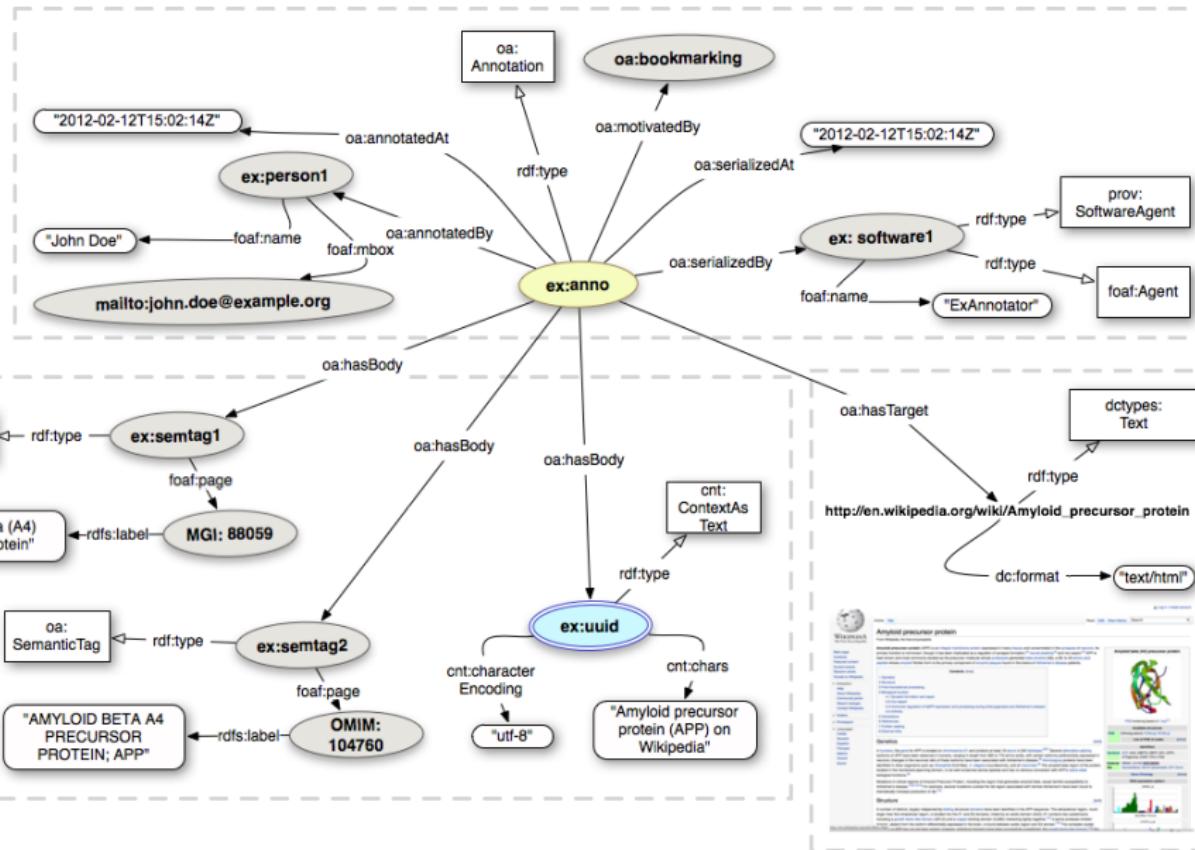
- ❑ structural storage of data within published documents;
- ❑ processing stored data computationally;
- ❑ integration of data structures and data objects globally.

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)

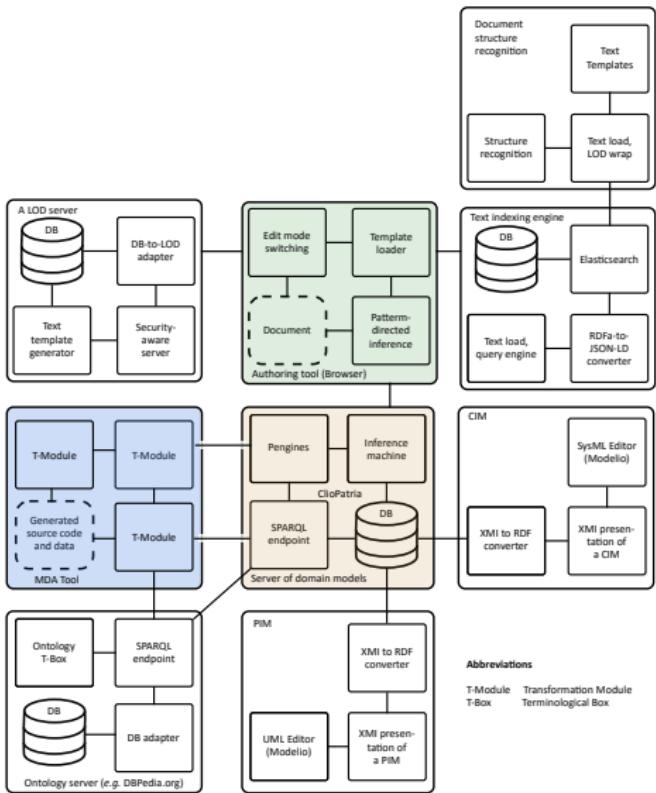


Representation

```
<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>
      . . . . .
    </article>
  </main>
</body>
```

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

Generated part of a study program

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, ...

запись,

- 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

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)
вы выпускника - бакалавр

Форма обучения - очная

Иркутск 2016 г.

Согласовано с УМК факультета (института)

Рекомендовано кафедрой:

Протокол № от " " 20__ г.

Протокол № от " " 20__ г.

Председатель _____
(подпись)

Зав. кафедрой _____
(подпись)

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)

Целью 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

Resume on Document Representation

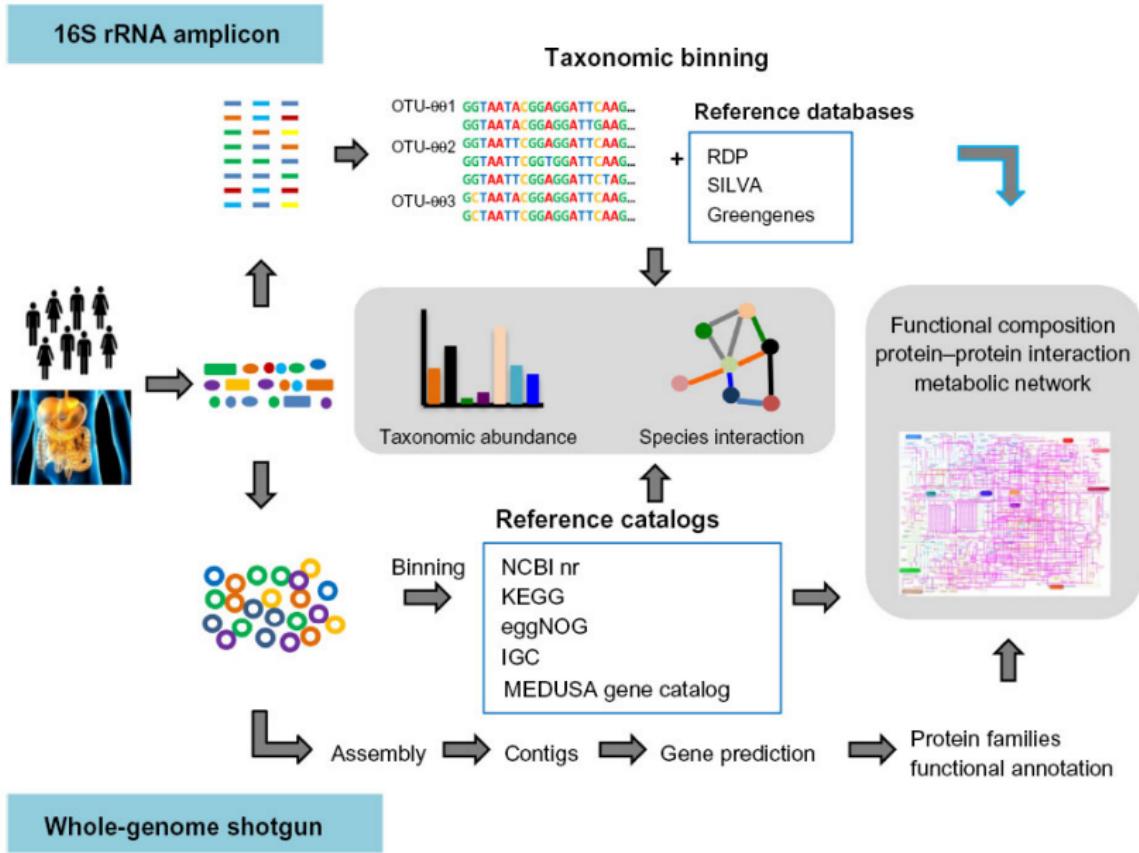
A tools (components) for digital archive implementation, which allows to device information systems and document processing services with the following features:

- ❑ 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.

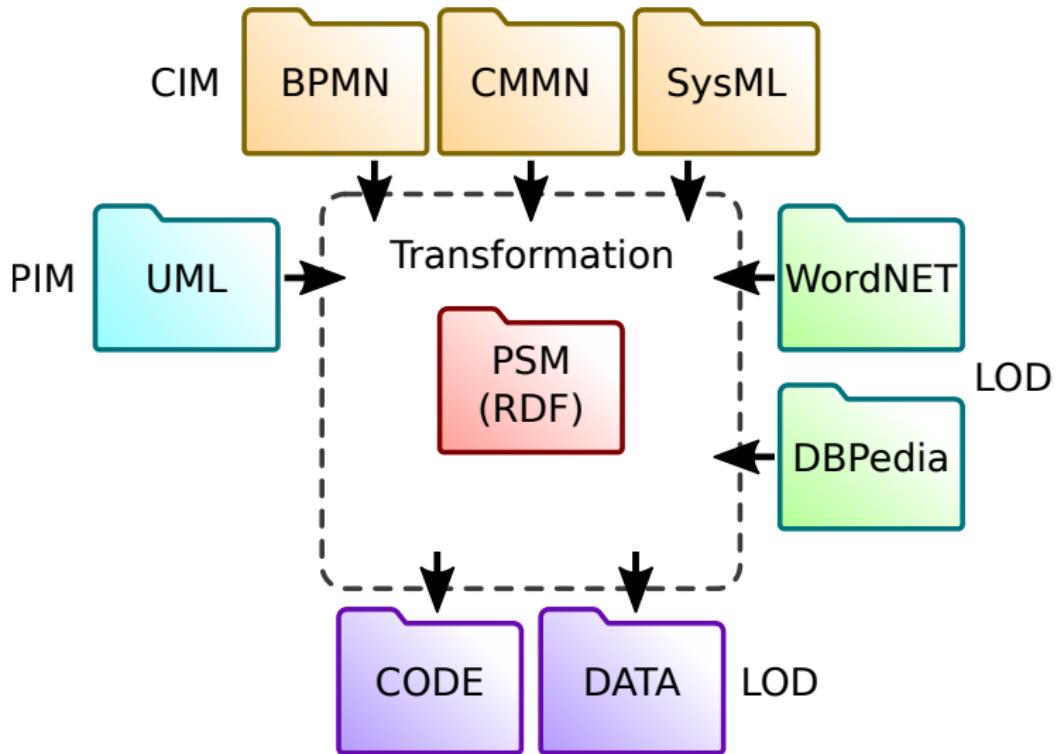
Applications

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

Microbiome study process

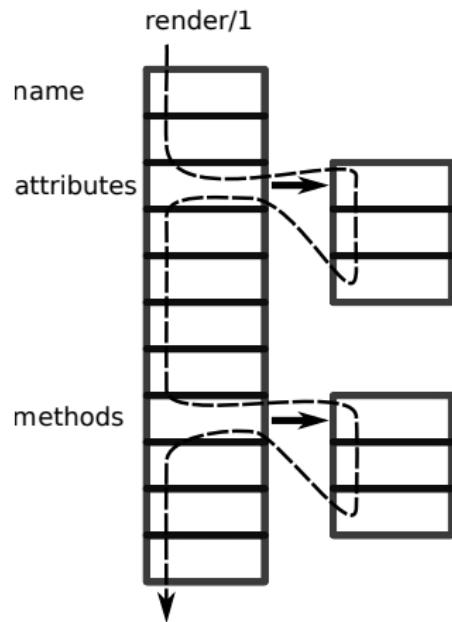


Model Driven Architecture and Linked Open Data



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::swritef(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.
```

1. Information is published in Internet with open access license;
2. It is represented in a machine-readable form, e.g., Excel table instead of a bitmap picture;
3. An open format used, e.g., CSV instead of Excel;
4. The format is based on W3C recommended standards, allowing RDF and SPARQL reference;
5. Published data refer to objects, forming context.

Thus, applications publish data as relations of objects (entities).

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(queryparam(_RDF,_Parameter),  
          extends(ngsquerybase)).  
  
    :- public(type/1).  
    type(Type) :-  
        ::attr(type, Type).  
    :- public(name/1).  
    name(Name) :- ::attr(dc:title, literal(Name)).  
    :- public(options/1).  
    options(Value) :- ::attr(options, Value).  
    :- public(options_default/1).  
    options_default(Value) :-  
        ::attr(optionsDefault, Value).  
% . . . . .  
    :- public(multiple_selection_allowed/0).  
    multiple_selection_allowed :-  
        ::bool_attr(multipleSelectionAllowed).  
    :- public(required/0).  
    required :-  
        ::bool_attr(required).  
    :- public(important/0).  
    important :-  
        ::bool_attr(important).  
    :- protected(attr/2).  
    attr(NS:Name, Value) :-  
        ::ngs(RDF),  
        ::second(Parameter),  
        rdf_db:rdf_global_object(Value, V),  
        RDF:rdf(Parameter, NS:Name, V).  
    attr(Name, Value) :-  
        \+ Name=_:*!,  
        ::ngs(RDF),  
        ::second(Parameter),  
        rdf_db:rdf_global_id(Value, V),  
        RDF:rdf(Parameter, ngsp:Name, V).  
% . . . .  
:- end_object.
```