# Logical Approach in Software and Data Design
# 软件和数据设计中的逻辑方法

Evgeny Cherkashin

Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences, Irkutsk, Russia
俄罗斯伊尔库茨克，Matrosov
俄罗斯科学院西伯利亚分院系统动力学与控制理论研究所
eugeneai@icc.ru

2024, March, Shandong, China



## QR-Code of the presentation



## Model–Driven Architecture: Research objectives

Main objective of the research is to construct a MDA technology based on nowadays system modeling visual languages (SysML, UML, BPMN, CMMN) and existing Semantic Web vocabularies and technologies. The following techniques and software are under development:

1. CIM representation with SysML, BPMN, CMMN, and results of source code processing,
2. CIM, PIM, PSM representation in UML, RDF with existing vocabularies,
3. transformation implementation with logical language Logtalk,
4. usage of LOD sources in transformations for obtaining additional semantic data,
5. generation of documents and user interfaces with LOD markup.



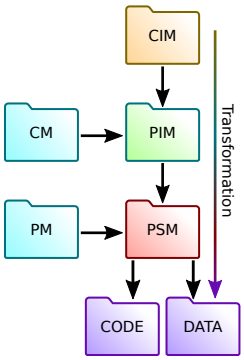
## 模型驱动架构： 研究目标

研究的主要目标是基于当今的系统建模可视化语言（SysML、UML、BPMN、CMMN）和现有的语义网词汇表和技术构建一种 MDA 技术 以下技术和软件正在开发中

1. 用 SysML、BPMN、CMMN 和源代码处理结果表示 CIM、
2. 用 UML、RDF 和现有词汇表表示 CIM、PIM 和 PSM、
3. 使用逻辑语言 Logtalk 实现转换、
4. 在转换中使用 LOD 源以获取额外的语义数据、
5. 使用 LOD 标记生成文档和用户界面。



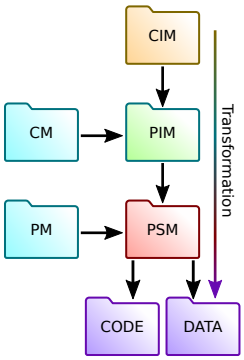
## Model–Driven Architecture




MDA 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.



## 模型驱动架构（Model–Driven Architecture）




MDA 模型驱动架构（Model–Driven Architecture）；
CIM Computationally Independent Model（独立计算模型）；
CM 计算模型；
PIM 平台独立模型；
PM 平台模型；
PSM 特定平台模型；
CODE 软件源代码；
DATA 初始数据库状态。



## Logtalk as transformation definition language

We have chosen Logtalk as it

- inherits widely known Prolog language syntax and runtime;
- is 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 various Prolog engines.

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



## Logtalk 作为转换定义语言

我们选择Logtalk是因为它

- 继承了广为人知的 Prolog 语言的语法和运行时；
- 作为宏包实现，性能损失约为1.5
- 具有灵活的语义：我们可以在相同的语法中定义转换和约束；
- 实现了面向对象的知识（规则）结构化、封装和替换；
- 实现转换的组合方式；
- Logtalk是一个强大的引擎，可以对对象到对象的消息（事件）发布约束；
- 为各种Prolog引擎提供实现。

这种常规语言允许我们使用它与 MDA 转换没有直接关系的库。

## Linked Open Data, LOD

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).



## Linked Open Data, 关联开放数据, LOD

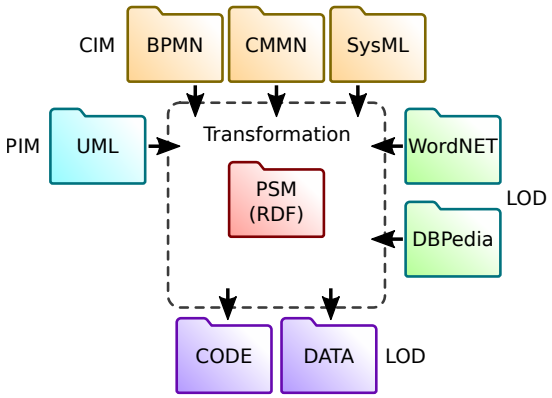
1. 信息在互联网上发布，采用开放式获取许可；
2. 它以机器可读的形式表示，例如 Excel 表格，而不是位图图片；
3. 使用的开放格式，如 CSV 而非 Excel；
4. 该格式基于 W3C 推荐的标准，允许引用 RDF 和 SPARQL；
5. 发布的数据指代对象，形成上下文。

因此，应用程序以对象（实体）关系的形式发布数据。



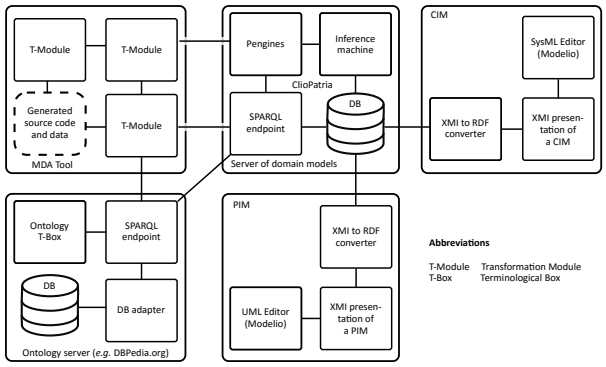
## Model Driven Architecture and Linked Open Data

模型驱动架构和关联开放数据



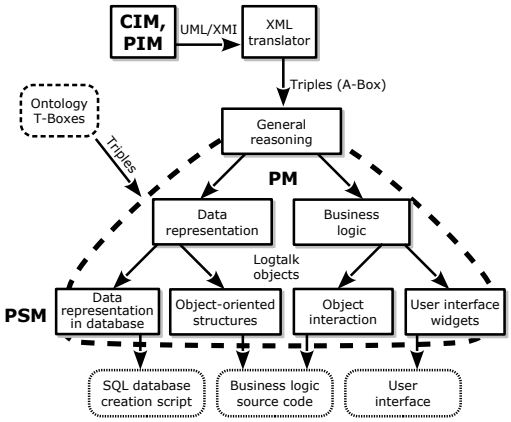



## MDA infrastructure, 基础设施





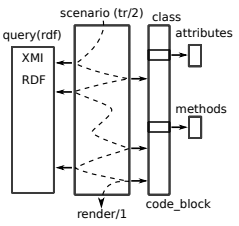

## Architecture of transformation modules, 转换模块的结构






## PSM: Scenario of a Class synthesis, PSM: 课堂综合情景

```
:- object(direct(_Package,_LocalProf,_CodeProf)).  % 转换驱动程序对象
:- public([tr/4,tr/3]).                            % 类综合方案的公共接口
% . . . . . .
tr(class, Class, ClassID):- ::package(Package),    % 合成一个类
   query(Package)::class(Name, ClassID),           % XMI 中的查询包结构
   create_object(Class,     % . . . .              % 创建类对象
   create_object(Attributes, % . . . .             % 创建属性对象
   create_object(Methods,   % . . . .              % ...方法
   Class::name(Name),                              % 为班级命名。
   % Generate attributes of the class,
   % organizing them in a local database.
   % ...methods...
   Class::attributes(Attributes),                  % 为类设置属性。
   Class::methods(Methods).                        % ...方法。

tr(attribute, Attribute, ClassID, AttributeID):-   % 属性转换
   ::package(Package),
   query(Package)::attribute(Name,ClassID,AttrID),
   create_object(Attribute, % . . . .
   Attribute::name(Name).                          % 为属性命名。

tr(method, Method, ClassID, MethodID):-            % 方法的转变
   ::package(Package),
   query(Package)::method(Name,ClassID,MethodID),
   create_object(Method,    % . . . .
   Method::name(Name).                             % 方法名称
:- end_object.
```





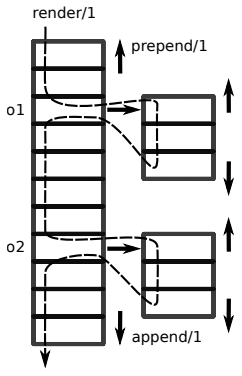

## Implementation of Query object, 实现查询对象

```
:- object(query(_XMI)).
:- protected(xmi/1).
:- public([class/2, attribute/3, method/3]).
xmi(XMI) :- parameter(1, XMI).
class(Name, ID):-                  % 识别 RDF 中的类别
   ::xmi(XMI),
   XMI::rdf(ID,rdf:type,uml:'Class'),
   XMI::rdf(ID,rdfs:label, literal(Name)).
attribute(Name, ClassID, ID):-     % ...属性...
   ::xmi(XMI),
   XMI::rdf(ClassID, xmi:ownedAttribute, ID),
   XMI::rdf(ID, rdfs:label, literal(Name)).
method(Name, ClassID, ID):-        % ...方法...
   ::xmi(XMI),
   XMI::rdf(ClassID, xmi:ownedOperation, ID),
   XMI::rdf(ID, rdfs:label, literal(Name)).
% . . . . . .
:- end_object.
```



## Code Block (idea is taken from llvmlite*, 代码块，创意来自 ”llvmlite ” 图书馆)

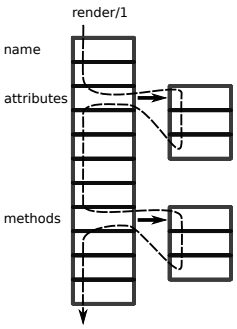
```
:- object(code_block, specializes(root)).
% Public interface of the object
:- public([append/1, prepend/1, clear/0,
   render/1, render_to/1, remove/1,
   item/1, items/1]).
% Code block items
:- dynamic([item_/1]).
:- private([item_/1]).
% Methods specialized during inheritance
:- protected([renderitem/2, 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

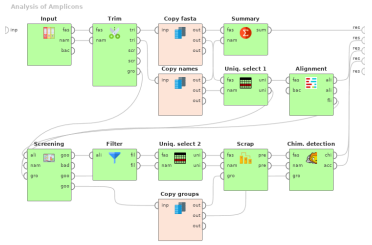
# PSM of a Python Class as a specialization of Code Block, 作为代码块特化的 Python 类

```
:- 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].
```






# Applications: Dataflow representation of NGS analysis of amplicons, 应用： 扩增子 NGS 分析的数据流表示法




| Term 术语 | Description 说明 |
|---|---|
| NGS | 新一代测序 |
| Amplicon | 复制多次的 DNA 或 RNA 部分 |
| Mothur | 用于 NGS 研究的软件工具集 |
| Rapidminer | 视觉分析工具 |

Green blocks are Mothur modules. Others are Rapidminer modules.
绿色块为 Mothur 模块。其他为 Rapidminer 模块。



# Rapidminer module synthesis, 模块合成

```
vector<string> AlignCommand::setParameters(){ // 模块源的一部分
try {
 CommandParameter ptemplate(" reference" ," InputTypes" ," " ," " ," none" ," none" ," none" ," " ,false,true,true); parameters.push_back(ptempla
 CommandParameter pcandidate(" fasta" ," InputTypes" ," " ," " ," none" ," none" ," none" ," fasta-alignreport-accnos" ,false,true,true); parameters
 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; // 生成的 Java 模块
// 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 List<ParameterType> getParameterTypes() {
  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);
 }
}
```



# RDF (TTL) representation and ad its query object, RDF 表示法并添加其查询对象

```
@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). % 立面对象
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).
% . . . . .
```



# Discussion

Interesting positive impressions obtained:

- Logtalk and RDF are flexible, sufficiently universal and convenient implementation infrastructures for MDA;
- The best implemenation 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;
- Prolog is not a popular language in MDA, neither Logtalk.



# Discussion, 讨论

有趣的正面印象:

- Logtalk 和 RDF 是灵活、足够通用和方便的 MDA 实现基础架构；
- 最好的实现手段是对规则进行Prolog谓词封装和Logtalk对象封装；
- 并非所有的Logtalk属性都得到了研究：在消息监视器的基础上，可能会开发出更复杂的编程技术。

技术问题使得这种方法有些困难：

- 非常简单的任务就需要花费太多精力，比如文本处理：将标识符转换成 CamelCase；
- 为了找到一个领域的词汇而上网太花时间了，但这比开发更有成效；
- Prolog在MDA中并不流行，Logtalk也不是。



# 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.



# Document authoring and storage, 文件编写和存储

在大多数情况下，创建文档的结果是

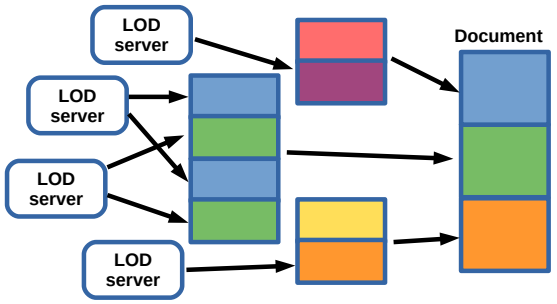
- 文本处理人员的创造性活动（创作）；
- 打印数字副本或数据库中的数据记录；
- 对数据库记录进行汇总操作（报告）。

然后
以实体纸张和/或数字文档（PDF、DOCX、HTML）的形式存储。
自2000年以来，语义网（Semantic Web）和关联开放数据（Linked Open Data，LOD）得到了发展，从而可以

- 在已发布的文档中以结构化方式存储数据；
- 计算处理存储的数据；
- 在全球范围内整合数据结构和数据对象。

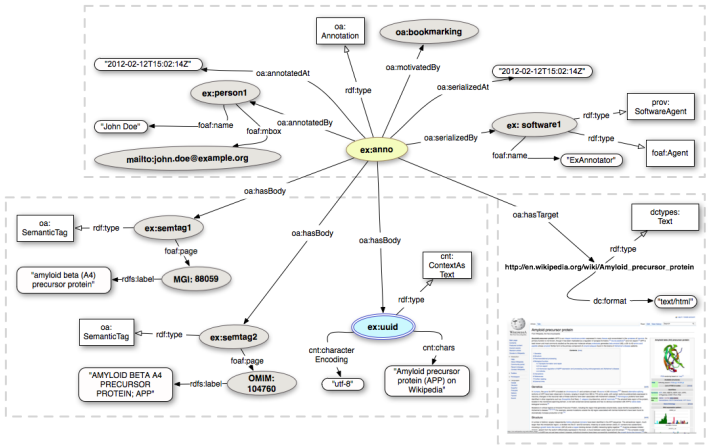
The aim of this research is to develop technologies, software and services allowing construction of digital archives supporting document data inclusion and inference from existing documents.
这项研究的目的是开发技术、软件和服务，以便构建数字档案，支持文件数据的包含和现有文件的推理。

## Structure of a document, 文件结构






## Open Annotation ontology (oa), 开放注释本体论



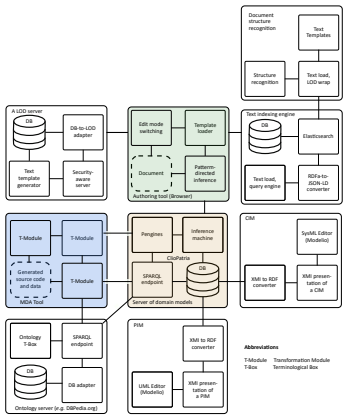



## Representation, 代表性

```
<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>
 . . . . . .
```



## Architecture, 建筑学






## Generated list of title page preambles, 生成扉页序言列表

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 course description, 生成课程说明的一部分

**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, seminars, …, 导入讲座和研讨会的时间分配、

загрузке;
- 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 |



## Complete document, 完整文件

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

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 № от " " 20 г. Протокол № от " " 20 г.

Председатель Зав. кафедрой
(подпись) (И.О. Ф.)

**Содержание**



**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

## Used ontologies

- Friend-of-a-friend (foaf) - agent information: individuals, legal entities, program agents.
- Provenance (prov) - references between documents.
- Dublin Core (dc) - edited annotation mark up.
- DBPedia resource (dbr) – references to instant objects and classes.
- Schema.org (schema) - Google, Yandex, Yahoo, etc. searchable objects, structural elements.
- The Bibliographic Ontology (bibo) - literature reference mark up.



## 使用的本体

- Friend-of-a-friend (foaf):
  代理人信息：个人、法人实体、计划代理人
- Provenance (prov): 文件之间的引用
- Dublin Core (dc): 编辑的注释标记
- DBPedia resource (dbr): 即时对象和类的引用
- Schema.org (schema): Google、Yandex、Yahoo 等搜索对象、结构元素
- The Bibliographic Ontology (bibo): 文献参考标注



## Conclusion

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.



## 结论

实施数字档案的工具（组件），可以 设备信息系统和文件处理服务具有以下功能:

- 加载 LOD 标注文件、提取、存储在图表中并为 RDF 数据编制索引
- 以三元组或全文搜索查询结果的形式检索 RDF 数据
- 利用基于浏览器的上下文推理机，将现有 LOD 数据及其内容动态结合到新文件中
- 使用服务器端推理机（Prolog）处理 RDF 数据。处理 RDF 数据
- 将创建的 RDFa 标记 HTML5 文档转换为 Excel 和 Word 格式。

Applications

- 文件编写自动化
- 根据上下文进行编辑
- 自组织全球文件流
- 文件作为信息系统的数据源



## TabbyXL

Software Platform for Rule-Based Spreadsheet Data Extraction and Transformation
基于规则的电子表格数据提取和转换软件平台

Alexey Shigarov, Vasiliy Khristyuk, et al.

shigarov@icc.ru



## Motivation

- About arbitrary spreadsheet tables
  - A large volume of valuable data for science and business applications
  - A big variety of layout, style, and content features
  - Human-centeredness (incorrect structure and messy content)
  - No explicit semantics for interpretation by computers
- Challenges
  - How to extract tables from worksheets
  - How to recognize and correct cell structure anomalies
  - How to recover semantics needed for the automatic interpretation
  - How to conceptualize extracted data by using external vocabularies



## 动机

- 关于任意电子表格表格
  - 为科学和商业应用提供大量宝贵数据
  - 丰富多样的布局、风格和内容功能
  - 以人为本（结构不正确，内容杂乱无章）
  - 没有明确的语义供计算机解释
- 挑战
  - 如何从工作表中提取表格
  - 如何识别和纠正细胞结构异常
  - 如何恢复自动解释所需的语义
  - 如何使用外部词汇表将提取的数据概念化



## Background

Table understanding includes the following tasks

1. Extraction, detecting a table and recognizing the physical structure of its cells
2. Role analysis, extracting functional data items from cell content
3. Structural analysis, recovering internal relationships between extracted functional data items
4. Interpretation, linking extracted functional data items with external vocabularies (general-purpose or domain-specific ontologies)

# 背景介绍

表格理解 包括以下任务

1. 提取: 检测表格并识别其单元格的物理结构
2. 角色分析: 从单元格内容中提取功能数据项
3. 结构分析: 恢复提取的功能数据项之间的内部关系
4. 口译: 将提取的功能数据项与外部词汇表（通用或特定领域本体）连接起来

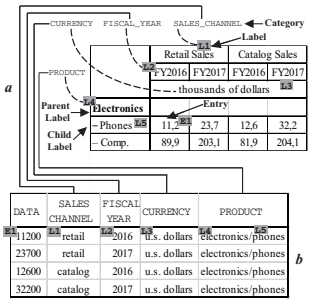


# Contribution

TabbyXL is a software platform aiming at the development and execution of rule-based programs for spreadsheet data extraction and transformation from arbitrary (a) to relational tables (b)

## Novelty

- Table object model assigning roles to data items, not cell
- CRL, domain-specific language to express user-defined rules for table analysis and interpretation
- CRL-to-Java translator to synthesize executable programs for spreadsheet data transformation



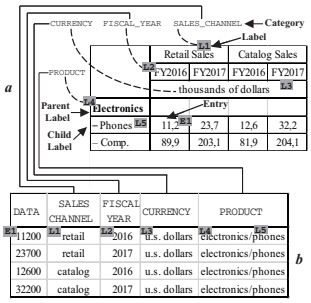



# 捐款

TabbyXL 是一个软件平台，旨在开发和执行基于规则的程序，用于电子表格数据提取和从任意表（a）到关系表（b）的转换。

## Novelty

- 为数据项而非单元格分配角色的表格对象模型
- CRL，特定领域语言，用于表达用户定义的表格分析和解释规则
- CRL 到 Java 翻译器，用于合成电子表格数据转换的可执行程序






# Table Object Model

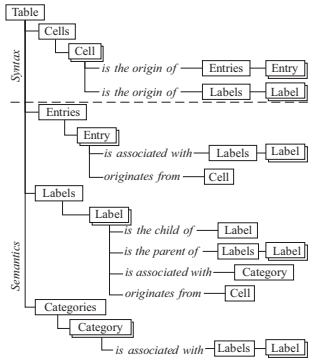
## Physical Layer

Cells characterized by layout, style, and content features

## Logical Layer

Functional data items and their relationships:

- entries (values)
- labels (keys)
- categories (concepts)
- entry-label pairs
- label-label pairs
- label-category pairs






# 表格对象模型

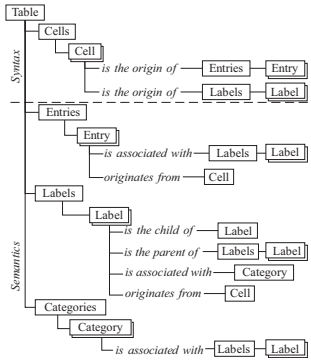
## 物理层

以布局、样式和内容为特征的单元格

## 逻辑层

功能数据项及其关系:

- 条目（数值）
- 标签（键）
- 类别（概念）
- 条目标签对
- 标签-标签对
- 标签-类别对






# CRL Grammar

```
rule       = 'rule' <a Java integer literal> 'when' condition
             'then' action 'end' <EOL> {rule} <EOF>
condition  = query identifier [':' constraint {',' constraint}
             [',' assignment {',' assignment}]] <EOL> {condition}
constraint = <a Java boolean expr>
assignment = identifier ':' <a valid Java expr>
query      = 'cell' | 'entry' | 'label' | 'category' | 'no cells' |
             'no entries' | 'no labels' | 'no categories'
action     = merge | split | set text | set indent | set mark |
             new entry | new label | add label | set parent |
             set category | group <EOL> {action}
merge      = 'merge' identifier 'with' identifier
split      = 'split' identifier
set text   = 'set text' <a Java string expr> 'to' identifier
set indent = 'set indent' <a Java integer expr> 'to' identifier
set mark   = 'set mark' <a Java string expr> 'to' identifier
new entry  = 'new entry' identifier ['as' <a Java string expr>]
new label  = 'new label' identifier ['as' <a Java string expr>]
add label  = 'add label' identifier | (<a Java string expr>
             'of' identifier | <a Java string expr>)
             'to' identifier
set parent = 'set parent' identifier 'to' identifier
set category = 'set category' identifier | <a Java string expr>
             'to' identifier
group      = 'group' identifier 'with' identifier
identifier = <a Java identifier>
```



# Cell Cleansing

The actions correct an inaccurate layout and content of a hand-coded table

- <merge> combines two adjacent cells when they share one border
- <split> divides a merged cell that spans $n$-tiles (row-column intersections) into $n$-cells
- <set text> modifies a textual content of a cell
- <set indent> modifies a text indentation of a cell

## Example

when
 cell corner: cl == 1, rt == 1, blank
 cell c: cl > corner.cr, rt > corner.rb
then
 split c



# 细胞清洁

这些操作可纠正手工编码表格中不准确的布局和内容

- <merge> 当相邻两个单元格共享一个边界时，将其合并
- <split> 将跨 $n$ 格（行列交叉点）的合并单元格划分为 $n$ 单元格
- <set text> 修改单元格的文本内容
- <set indent> 修改单元格的文本缩进

## Example

when
 cell corner: cl == 1, rt == 1, blank
 cell c: cl > corner.cr, rt > corner.rb
then
 split c

# Role Analysis

The actions recover entries and labels as functional data items presented in a table

- <set mark> annotates a cell with a user-defined tag that can be used in subsequent table analysis
- <new entry> (<new label>) creates an entry (label) from a cell content with the use of an optional string processing

**Example**

```
when
 cell corner: cl == 1, rt == 1, blank
 cell c: cl > corner.cr, rt > corner.rb
then
 new entry c
```



# 角色分析

操作恢复作为功能数据项的条目和标签，以表格形式呈现

- <set mark>
  为单元格标注用户定义的标签，该标签可用于后续表格分析
- <new entry> (<new label>)
  使用可选的字符串处理，从单元格内容创建条目（标签）。

**Example**

```
when
 cell corner: cl == 1, rt == 1, blank
 cell c: cl > corner.cr, rt > corner.rb
then
 new entry c
```



# Structural Analysis

The actions recover pairs of two kinds: entry-label and label-label

- <add label> associates an entry with a label
- <set parent> binds two labels as a parent and its child

**Example**

```
when
 cell c1: cl == 1
 cell c2: cl == 1, rt > c1.rt, indent == c1.indent + 2
 no cells: cl == 1, rt > $c1.rt, rt < $c2.rt, indent == $c1.indent
then
 set parent c1.label to c2.label
```



# 结构分析

恢复的操作对有两种：条目-标签和标签-标签

- <add label> 将条目与标签关联
- <set parent> 将两个标签绑定为父标签和子标签

**Example**

```
when
 cell c1: cl == 1
 cell c2: cl == 1, rt > c1.rt, indent == c1.indent + 2
 no cells: cl == 1, rt > $c1.rt, rt < $c2.rt, indent == $c1.indent
then
 set parent c1.label to c2.label
```



# Interpretation

The actions serve to recover label-category pairs

- <set category> associates a label with a category
- <group> places two labels to one group that can be considered as an undefined category

**Example**

```
when
 label l1: cell.mark == ” stub”
 label l2: cell.mark == ” stub”, cell.rt == l1.cell.rt
then
 group l1 with l2
```



# 口译

这些操作有助于恢复标签-类别对

- <set category> 将标签与类别关联
- <group> 将两个标签贴在一个可视为未定义类别的组上

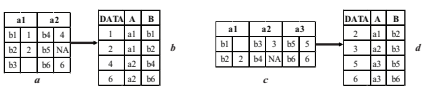
**Example**

```
when
 label l1: cell.mark == ” stub”
 label l2: cell.mark == ” stub”, cell.rt == l1.cell.rt
then
 group l1 with l2
```



# Illustrative Example

The transformation of arbitrary tables with the same layout features (a and c) to their canonical versions (b and d)



The ruleset for the cell cleansing (a), role analysis (b, c), structural analysis (d, e), and interpretation (f, g)

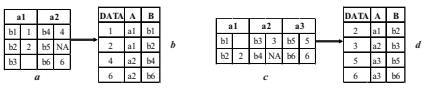
```
a  when cell c: c.text.matches("NA")
   then set text "" to c

b  when cell c: (cl % 2) == 0, !blank
   then new entry c

c  when cell c: (cl % 2) == 1
   then new label c

d  when
     entry e
     label l: cell.cr == e.cell.cr
   then add label l to e

e  when
     entry e
     label l: cell.rt == e.cell.rt, cell.cl == e.cell.cl - 1
   then add label l to e

f  when label l: cell.rt == 1
   then set category "A" to l

g  when label l: cell.rt > 1
   then set category "B" to l
```

This example is reproducible at https://codeocean.com/capsule/5326436



# 示例

将具有相同布局特征的任意表格（(a) 和 (c)）转换为其规范版本（(b) 和 (d)

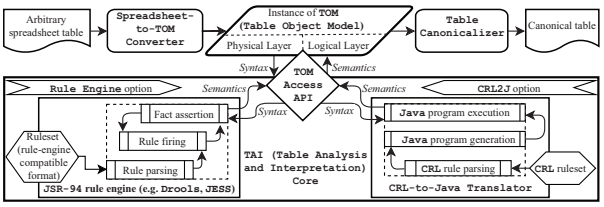


细胞清洗的规则集 (a), 角色分析 (b, c), 结构分析 (d, e), 和解释 (f, g)

```
a  when cell c: c.text.matches("NA")
   then set text "" to c

b  when cell c: (cl % 2) == 0, !blank
   then new entry c

c  when cell c: (cl % 2) == 1
   then new label c

d  when
     entry e
     label l: cell.cr == e.cell.cr
   then add label l to e

e  when
     entry e
     label l: cell.rt == e.cell.rt, cell.cl == e.cell.cl - 1
   then add label l to e

f  when label l: cell.rt == 1
   then set category "A" to l

g  when label l: cell.rt > 1
   then set category "B" to l
```

该示例可在以下网址复制 https://codeocean.com/capsule/5326436

# Architecture




Two options are provided

**Rule Engine option**

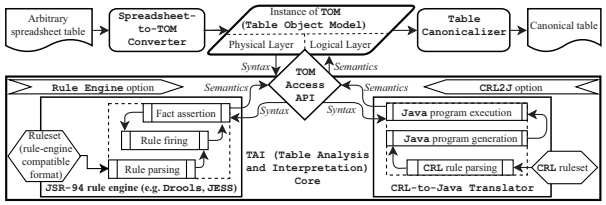
Executing a ruleset in an appropriate format with a JSR-94 compatible rule engine (e.g. Drools, Jess)

**CRL2J option**

Translating a ruleset expressed in CRL to an executable Java program



# 建筑学




提供两种选择

**规则引擎选项**

使用与 JSR-94 兼容的规则引擎（如 Drools、Jess）以适当格式执行规则集

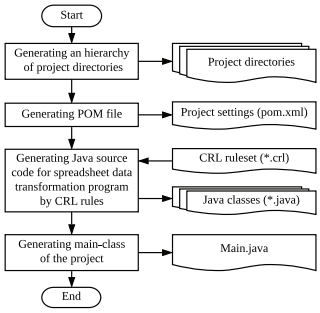
**CRL2J 选择权**

将 CRL 表达的规则集转换为可执行 Java 程序

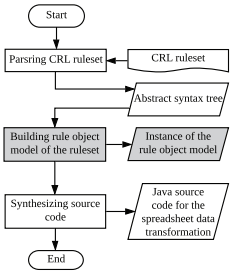


# CRL2J Translation

Workflow for generating a Maven-project of a spreadsheet data transformation program



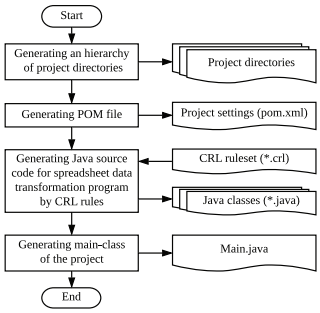

Workflow for translating a CRL ruleset to Java source code



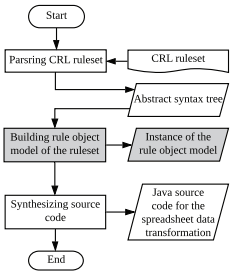



# CRL2J 翻译

生成电子表格数据转换程序 Maven 项目的工作流程



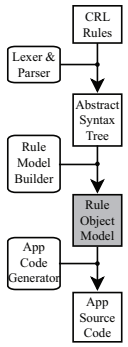

将 CRL 规则集转换为 Java 源代码的工作流程



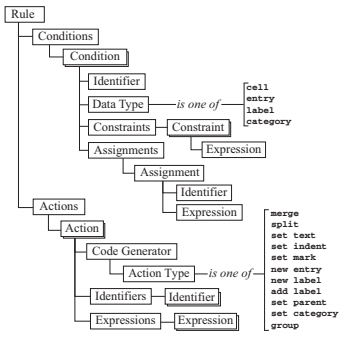



# CRL2J Translation

In the Workflow



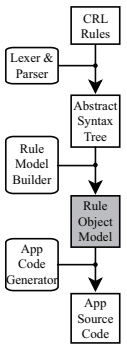

Rule Object Model



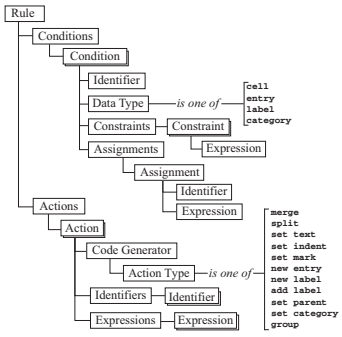



# CRL2J 翻译

在工作流程中




规则对象模型






# CRL2J Translation

**Example (Source Rule)**

```
when
 cell corner: cl == 1, rt == 1, blank
 cell c: cl > corner.cr, rt > corner.rb, ! marked
then
 set mark ” @entry ” to c
 new entry c
```

**Example (Fragment of the Generated Java Code)**

```
...
Iterator<CCell> iterator1 = getTable().getCells();
while (iterator1.hasNext()) {
 corner = iterator1.next();
 if ((corner.getCl() == 1) && (corner.getRt() == 1) && ...
  Iterator<CCell> iterator2 = getTable().getCells();
  while (iterator2.hasNext()) {
...
```



# CRL2J 翻译

**Example (来源规则)**

```
when
 cell corner: cl == 1, rt == 1, blank
 cell c: cl > corner.cr, rt > corner.rb, ! marked
then
 set mark ” @entry ” to c
 new entry c
```

**Example (生成的 Java 代码片段)**

```
...
Iterator<CCell> iterator1 = getTable().getCells();
while (iterator1.hasNext()) {
 corner = iterator1.next();
 if ((corner.getCl() == 1) && (corner.getRt() == 1) && ...
  Iterator<CCell> iterator2 = getTable().getCells();
  while (iterator2.hasNext()) {
...
```

# Performance Evaluation

The results of the transformation of 200 tables of Troy200 dataset

| Metrics | Role analysis | | Structural analysis | |
|---|---|---|---|---|
| | Type of instances | | | |
| | entries | labels | entry-label pairs | label-label pairs |
| Recall | 0.9813 $\frac{16602}{16918}$ | 0.9965 $\frac{4842}{4859}$ | 0.9773 $\frac{34270}{35066}$ | 0.9389 $\frac{1951}{2078}$ |
| Precision | 0.9996 $\frac{16602}{16609}$ | 0.9364 $\frac{4842}{5171}$ | 0.9965 $\frac{34270}{34389}$ | 0.9784 $\frac{1951}{1994}$ |
| $F$-score | 0.9904 | 0.9655 | 0.9868 | 0.9582 |

## Metrics

$$\text{recall} = \frac{|R \cap S|}{|S|} \quad \text{precision} = \frac{|R \cap S|}{|R|}$$

$S$ is a set of instances in a source table, $R$ is a set of instances in its canonical form

All data and steps to reproduce the results are available at http://dx.doi.org/10.17632/ydcr7mcrtp.5



# 绩效评估

Troy200 数据集 200 个表格的转换结果

| 衡量标准 | 角色分析 | | 结构分析 | |
|---|---|---|---|---|
| | 实例类型 | | | |
| | 参赛 | 标签 | 条目标签对 | 标签-标签对 |
| 回顾 | 0.9813 $\frac{16602}{16918}$ | 0.9965 $\frac{4842}{4859}$ | 0.9773 $\frac{34270}{35066}$ | 0.9389 $\frac{1951}{2078}$ |
| 精确度 | 0.9996 $\frac{16602}{16609}$ | 0.9364 $\frac{4842}{5171}$ | 0.9965 $\frac{34270}{34389}$ | 0.9784 $\frac{1951}{1994}$ |
| $F$-总谱 | 0.9904 | 0.9655 | 0.9868 | 0.9582 |

## Metrics

$$\text{回顾} = \frac{|R \cap S|}{|S|} \quad \text{精确度} = \frac{|R \cap S|}{|R|}$$

$S$ 是源表中的一组实例, $R$ 是一组典型形式的实例

所有数据和重现结果的步骤可从以下网址获取 http://dx.doi.org/10.17632/ydcr7mcrtp.5



# Performance Evaluation

The comparison of the running time by using TabbyXL with three different options for transforming 200 tables of Troy200 dataset

| Running time of | CRL2J | Drools | Jess |
|---|---|---|---|
| Ruleset preparation ($t_1$) | 2108* ms | $1711^{\dagger}$ ms | $432^{\dagger}$ ms |
| Ruleset execution ($t_2$) | 367** ms | $1974^{\ddagger}$ ms | $4149^{\ddagger}$ ms |

* $t_1$ — a time of parsing and compiling the original ruleset into a Java program
** $t_2$ — a time of executing the generated Java program

$^{\dagger}$ $t_1$ — a time of parsing the original ruleset and adding the result into a rule engine session
$^{\ddagger}$ $t_2$ — a time of asserting facts into the working memory and matching rules against the facts

For testing, we used 3.2 GHz 4-core CPU



# 绩效评估

使用 TabbyXL 对 Troy200 数据集的 200 张表格进行转换时，三种不同选项的运行时间比较

| 运行时间 | CRL2J | Drools | Jess |
|---|---|---|---|
| 规则集准备 ($t_1$) | 2108* ms | $1711^{\dagger}$ ms | $432^{\dagger}$ ms |
| 规则集执行 ($t_2$) | 367** ms | $1974^{\ddagger}$ ms | $4149^{\ddagger}$ ms |

* $t_1$ — 解析原始规则集并将其编译成 Java 程序的时间
** $t_2$ — 执行生成的 Java 程序的时间

$^{\dagger}$ $t_1$ — 解析原始规则集并将结果添加到规则引擎会话的时间
$^{\ddagger}$ $t_2$ — 在工作记忆中断言事实并将规则与事实相匹配的时间

在测试中，我们使用了主频为 3.2 GHz 的 4 核 CPU



# Comparison with Others

## Role Analysis

- Contest task: The segmentation of a table into typical functional cell regions
- Testing dataset: Troy200
- Contestant: MIPS (TANGO)
- Accuracy: MIPS (TANGO) — 0.9899 vs. TabbyXL — 0.9950

## Structural Analysis

- Contest task: The extraction of header hierarchies from tables
- Testing dataset: A random subset of SAUS[a]
- Contestant: Senbazuru
- $F$-score: Senbazuru — 0.8860 vs. TabbyXL — 0.8657

[a] http://dbgroup.eecs.umich.edu/project/sheets/datasets.html



# 与他人比较

## 角色分析

- 竞赛任务: 将表格划分为典型的功能单元区域
- 测试数据集: Troy200
- 参赛者: MIPS (TANGO)
- 准确性: MIPS (TANGO) — 0.9899 vs. TabbyXL — 0.9950

## Structural Analysis

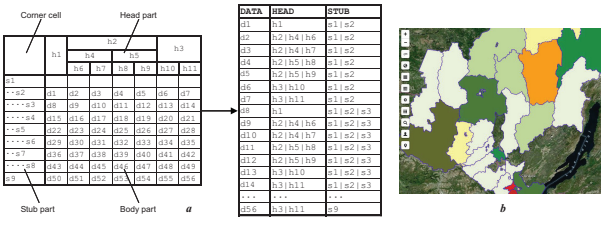
- 竞赛任务: 从表格中提取表头层次结构
- 测试数据集: 的一个随机子集 SAUS[a]
- 参赛者: Senbazuru
- $F$-总谱: Senbazuru — 0.8860 vs. TabbyXL — 0.8657

[a] http://dbgroup.eecs.umich.edu/project/sheets/datasets.html



# Application Experience

Populating a web-based statistical atlas of the Irkutsk region — (b) via extracting data from government statistical reports — (a)

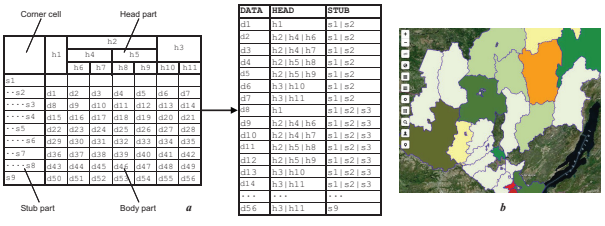



The more detail can be found at https://github.com/tabbydoc/tabbyxl/wiki/statistical-atlas



# 应用体验

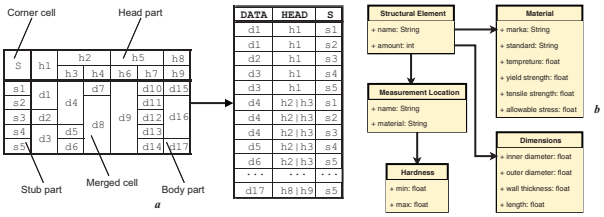
绘制伊尔库茨克州网络统计地图集 — (b)
通过从政府统计报告中提取数据 — (a)




更多详情请访问 https://github.com/tabbydoc/tabbyxl/wiki/statistical-atlas

# Application Experience

Generating conceptual models — (b) from arbitrary tables presented in industrial safety inspection reports — (a)

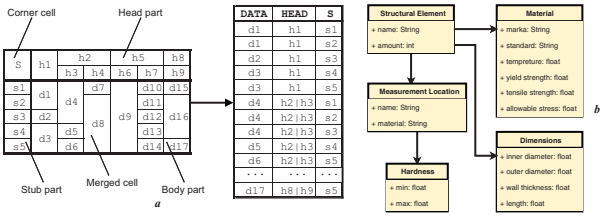



The more detail can be found at https://github.com/tabbydoc/tabbyxl/wiki/industrial-safety-inspection



# 应用体验

生成概念模型 — (b) 来自工业安全检查报告中的任意表格 — (a)




更多详情请访问 https://github.com/tabbydoc/tabbyxl/wiki/industrial-safety-inspection



# Conclusions & Further Work

- Impact on software development for spreadsheet data management
  - Table object model associating functional roles with data items
  - Table analysis and interpretation driven by user-defined rules
  - Formulated actions to recover missing semantics of arbitrary tables
  - Translation of rules to executable spreadsheet transformation programs
- Limitations
  - The inaccurate cell structure prevents the table analysis
  - The very limited interpretation (without external vocabularies)
- Further work
  - Rearrangement of cell structure by using visual (human-readable) cells
  - Detecting derived data by spreadsheet formulas
  - Enriching the table analysis by named entity recognition
  - Linking extracted data items with LOD cloud



# 结论和下一步工作

- 对电子表格数据管理软件开发的影响
  - 将功能角色与数据项关联起来的表对象模型
  - 根据用户定义的规则进行表格分析和解释
  - 为恢复任意表的缺失语义而制定的行动
  - 将规则转化为可执行的电子表格转换程序
- 局限性
  - 不准确的单元格结构妨碍了表格分析
  - 非常有限的解释（没有外部词汇表）
- 进一步的工作
  - 利用可视（人类可读）细胞重新排列细胞结构
  - 通过电子表格公式检测派生数据
  - 通过命名实体识别丰富表格分析
  - 将提取的数据项与 LOD 云连接起来



# Thanks!

Read more about the project at
http://td.icc.ru

The project source code is available at
https://github.com/tabbydoc/tabbyxl

But it is not all ...



# 谢谢!

有关该项目的更多信息，请访问
http://td.icc.ru

项目源代码见
https://github.com/tabbydoc/tabbyxl

但这并不是全部 ...



# Domain Knowledge Graphs Induction from Tables

Tables are the most available sources of information. They are valuable data sources for Knowledge Bases (KB)

Knowledge Base Construction Populating with document and structured table extracted data

Knowledge Base Population Populating with recognized new facts on entities from big text corpses

Knowledge base Augmentation Populating with relations with table data.


1. (Ré, 2014) Ré C., et al. Feature engineering for knowledge base construction. IEEE Data Eng. Bull., 37, 26–40, (2014).
2. (Balog, 2018) Balog K. Populating knowledge bases. Entity-Oriented Search. INRE, 39, 189–222, (2018).
3. (Zhang & Balog, 2020) Zhang S. & Balog K. Web table extraction, retrieval, and augmentation: A survey. ACM Trans. Intell. Syst. Technol., 11, (2020).




# 领域知识图谱 表格归纳法

表格是最常用的信息来源。它们是知识库（KB）的重要数据源

知识库建设 填充文件和结构化表格提取的数据

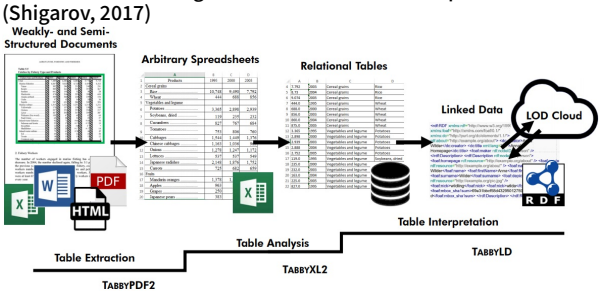
知识库人口 从大文本尸体中填充关于实体的公认新事实

知识库扩充 用表格数据填充关系


1. (Ré, 2014) Ré C., et al. Feature engineering for knowledge base construction. IEEE Data Eng. Bull., 37, 26–40, (2014).
2. (Balog, 2018) Balog K. Populating knowledge bases. Entity-Oriented Search. INRE, 39, 189–222, (2018).
3. (Zhang & Balog, 2020) Zhang S. & Balog K. Web table extraction, retrieval, and augmentation: A survey. ACM Trans. Intell. Syst. Technol., 11, (2020).

# Automatic Table Interpretation

There three main stages of Automatic table interpretation (Shigarov, 2017)

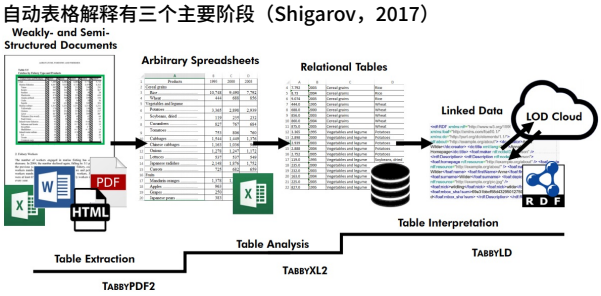



1. (Shigarov, 2017) Shigarov A., Mikhailov A. Rule-based spreadsheet data transformation from arbitrary to relational tables. Information Systems, 71, 123-136 (2017).



# 自动释表

自动表格解释有三个主要阶段（Shigarov，2017）



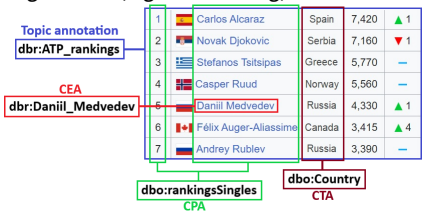

1. (Shigarov, 2017) Shigarov A., Mikhailov A. Rule-based spreadsheet data transformation from arbitrary to relational tables. Information Systems, 71, 123-136 (2017).



# Semantic Table Interpretation

Semantic interpretation (Annotation) of tables (Semantic Table Interpretation, STI) is a recognition of mutual and external relations between elements of table content. Externals relate to an enterprise KG and/or a global KG (e.g. DBPedia.org).

- Cell-Entity Annotation (CEA)
- Column-Type Annotation (CTA)
- Column Property Annotation (CPA)
- Topic Annotation

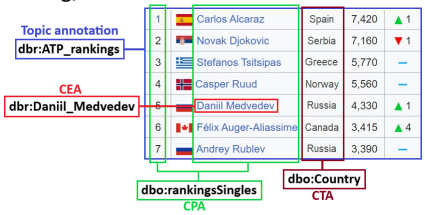





# Semantic Table Interpretation

表格的语义解释（注释）(Semantic Table Interpretation，STI)
是对表格内容元素之间相互关系和外部关系的一种确认。
外部关系涉及企业 KG
和/或全局 KG (例如 DBPedia.org)。

- 细胞实体注释（CEA）
- 列式注释 (CTA)
- 列属性注释 (CPA)
- 主题注释






# Cell-Entity Annotation

CEA comprises tho sequential steps as follows:

1. Select a candidate entity set from DBPedia.org for each value of a cell via SPARQL endpoint and DBPedia lookup.
2. Disambiguation

A SPARQL-query matching words of a phrase.

```
SELECT DISTINCT (str(?subject) as ?subject)
WHERE {
  ?subject a ?type .
  ?subject rdfs:label ?label .
  ?label <bif:contains> " .*%value1*." AND " .*%value2*." ... .
  FILTER NOT EXISTS { ?subject dbo:wikiPageRedirects ?r2 } .
  FILTER (!strstarts(str(?subject), " http://dbpedia.org/resource/Category:" )) .
  FILTER (!strstarts(str(?subject), " http://dbpedia.org/property/" )) .
  FILTER (!strstarts(str(?subject), " http://dbpedia.org/ontology/" )) .
  FILTER (strstarts(str(?type), " http://dbpedia.org/ontology/" )) .
  FILTER (lang(?label) = " en" )
}
ORDER BY ASC(strlen(?label))
LIMIT 100
```



# 细胞实体注释

CEA 包括以下三个连续步骤：

1. 通过 SPARQL 端点和 DBPedia 查找，为单元格的每个值从 DBPedia.org 中选择一个候选实体集。
2. 消歧义

匹配短语单词的 SPARQL 查询。

```
SELECT DISTINCT (str(?subject) as ?subject)
WHERE {
  ?subject a ?type .
  ?subject rdfs:label ?label .
  ?label <bif:contains> " .*%value1*." AND " .*%value2*." ... .
  FILTER NOT EXISTS { ?subject dbo:wikiPageRedirects ?r2 } .
  FILTER (!strstarts(str(?subject), " http://dbpedia.org/resource/Category:" )) .
  FILTER (!strstarts(str(?subject), " http://dbpedia.org/property/" )) .
  FILTER (!strstarts(str(?subject), " http://dbpedia.org/ontology/" )) .
  FILTER (strstarts(str(?type), " http://dbpedia.org/ontology/" )) .
  FILTER (lang(?label) = " en" )
}
ORDER BY ASC(strlen(?label))
LIMIT 100
```



# Evaluation on Test Table Sets

A well-known precision measurement (accuracy) is used for assessment

$$\text{Accuracy} = \frac{CC}{NC},$$

where $CC$ is the number of the correctly related columns to a categorical entity, and $CN$ is the total number of columns.

| Recognition stage | T2Dv2 | Tough_ Tables | Git-Tables |
|---|---|---|---|
| Stage 2, Atomic column classification | 0.994 | 0.956 | 0.938 |
| Stage 3, Column entity identification | 0.924 | – | – |

Comparison with analogs

| | TAIPAN | Table-Miner+ | T2Dv2 | Mantis-Table |
|---|---|---|---|---|
| Column entity identification | 0.540 | 0.871 | 0.924 | 0.979 |



# Evaluation on Test Table Sets

采用众所周知的精确测量（精度）进行评估

$$\text{准确性} = \frac{CC}{NC},$$

其中，$CC$ 是与分类实体正确相关的列数，$CN$ 是列的总数。

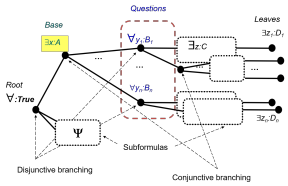
| 认可阶段 | T2Dv2 | Tough_ Tables | Git-Tables |
|---|---|---|---|
| 第 2 阶段，原子柱分类 | 0.994 | 0.956 | 0.938 |
| 第 3 阶段，列实体识别 | 0.924 | – | – |

与类似物的比较

| | TAIPAN | Table-Miner+ | T2Dv2 | Mantis-Table |
|---|---|---|---|---|
| 列实体标识 | 0.540 | 0.871 | 0.924 | 0.979 |

# Knowledge Representation and Reasoning: the PCF-Calculus

The main properties of the language of positively constructed formulas (PCF) and its calculi:

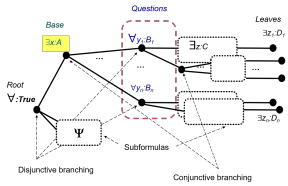



- PCFs have a large-block structure (tree-like) and consist of only positive quantifiers ∃ and ∀
- the PCF-based calculus have a unique inference rule
  - the proof in the PCF-calculus is organized as a question-answering procedure
  - PCF-calculus is both machine-oriented and human-oriented; it is compatible with heuristics
  - the semantic of the PCF-calculus can be changed without modifying axioms and the inference rule



# 知识表示与推理：PCF 微积分

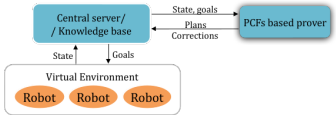
正构造公式（PCF）语言及其计算器的主要特性：

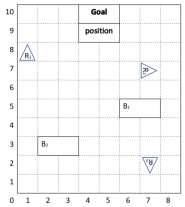



- PCF 具有大块结构（树状），仅由正量词∃ 和 ∀ 组成。
- 基于 PCF 的微积分有一个唯一的推理规则
  - PCF 微积分中的证明是以问题解答过程的形式组织的
  - PCF 微积分既面向机器，也面向人类；它与启发式方法兼容
  - 可以在不修改公理和推理规则的情况下改变 PCF 微积分的语义



# PCF-Based Method for Problem Solving

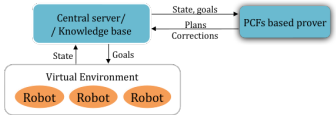


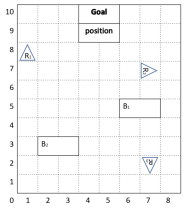




- The goal of the team of robot is to transport blocks to the target area
- Each block can be dragged by two or more robots
- The current state of the World and the goal of the group are formalized in PCF
- The PCF-based prover and a selection mechanism produce the optimal joint plan of actions for the team
- The current plan can be easily modified whenever the state of the World is changed



# 基于 PCF 的问题解决方法




- 机器人团队的目标是将积木运送到目标区域
- 每个积木可由两个或多个机器人拖动
- 世界的当前状态和小组的目标在 PCF 中形式化为
- 基于 PCF 的求证器和选择机制为团队生成最优的联合行动计划
- 只要”世界”的状态发生变化，就可以轻松修改当前计划



# A Master Degree Program. Semantic Technologies and Multiagent Systems

It is a joint effort of Saint-Petersburg Electrotechnical University (LETI), Irkutsk State University, and ISDCT SB RAS. Main subjects.

- Computation Geometry, Digital Signal Processing, Internet of Things,
- Semantic web, Semantic web Information System Development,
- AI Basics, Knowledge representation, Object-oriented Logic Programming,
- Answer Set Programming (SAT), Natural Language Processing,
- Machine Learning, Neural Networks, Deep Learning,
- Multiagent Systems, Optimization with Multiagent Systems.

Started at 2022-09-01.
https://etu.ru/sveden/education/programs/semanticheskie-tehnologii-i-mnogoagentnye-sistemy-01.04.02.html



# A Master Degree Program. 语义技术与多代理系统

该课程由圣彼得堡电工技术大学（LETI）、伊尔库茨克国立大学（Irkutsk State University）和俄罗斯科学院空间技术研究所（ISDCT SB RAS）联合开设。
主要课题。

- 计算几何、数字信号处理、物联网、
- 语义网、语义网信息系统开发、
- 人工智能基础、知识表示、面向对象逻辑编程、
- 答案集编程 (SAT)、自然语言处理、
- 机器学习、神经网络、深度学习、
- 多代理系统，多代理系统优化。

始于 2022-09-01。
https://etu.ru/sveden/education/programs/semanticheskie-tehnologii-i-mnogoagentnye-sistemy-01.04.02.html



# Conclusion (the final one)

- Classic knowledge-based systems are powerful AI tools for solving wide class of recognition problems and synthesis of various kind: source code, data objects, control
- Contemporary means combine classic and new approaches
- Less dependent on computational resources (as compared to machine learning)
- Allow justification of the produced solutions
- Cover a larger set of tasks
- Natural for math science, and require higher level of AI education



# 结论（最后）

- 经典的基于知识的系统是强大的人工智能工具，可用于解决广泛的识别问题和各种综合问题：源代码、数据对象、控制、数据处理、数据分析、数据挖掘。
- 当代手段结合了经典方法和新方法
- 较少依赖计算资源（与机器学习相比）
- 允许对所产生的解决方案进行论证
- 涵盖更多任务
- 自然适用于数学科学，需要更高水平的人工智能教育

Thank You!
谢谢大家!