Knowledge graph based distributed infrastructure for processing documents used for organizing education process

Evgeny A. Cherkashin, Victoria A. Popova

Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences, Irkutsk, Russia
Institute of Mathematics and Information Technologies, Irkutsk State University, Irkutsk, Russia
eugeneai@icc.ru, victorypopova1@gmail.com

AIIT'2022, October, 14, 2022
Zrenjanin, Serbia

Research and Development objectives

Irkutsk state university (ISU) has quite normal (required) level of automation in the areas of

- accounting ("1C:Accounting for budget institutions")
- education process planning ("1C:University")
- learning management, student state control ("Moodle")
- library data access with library information system (LMS "Irbis-64")

BUT other problems' automation has an **island character**.

- institutes of ISU develop software for local purposes
- do not share results between ISU community
- some solutions are implemented by a subdepartment IMIT of ISU on a request

Main objective of the present research is to creative activities of the faculty

- authoring a course program (CP),
- organizing processes, monitoring and control
- form a basis of educational process modeling to support the
  - ministry requirements compliance checking
  - compliance to domain of courses
  - individual education trajectories of students

Course plan authoring

One of the challenging problem is course documentation, such as CPs, mediation the previous version with the pan of education (EP). For CPs, following steps are to be performed:

1. Find a CP source at user's PC
2. Analyze actual EP, and find education unit (EU) distribution data, print/write it down
3. Recall the scenario of course teaching, add/remove/**comment** topics and laboratory work (LW) task set
4. Reconcile topics LW with exams question set and the set of competence
5. Fill in the results in the current template, upload **DOCX** to institute's cloud storage

Faculty member set varies, set of courses varies, EU distribution varies, ..., sources lost.

Schedule compilation and student progress monitoring

Class scheduling and student progress monitoring have many aspects of consideration

- a classic combinatorial optimization constraint satisfaction problem
- integration with education process planning software ("1C:University")
- Reconciling with other institutions occupying the same resources
- Accounting faculty requests and students' load

Long term result would be a **model of the education process**.

Assets, requirements

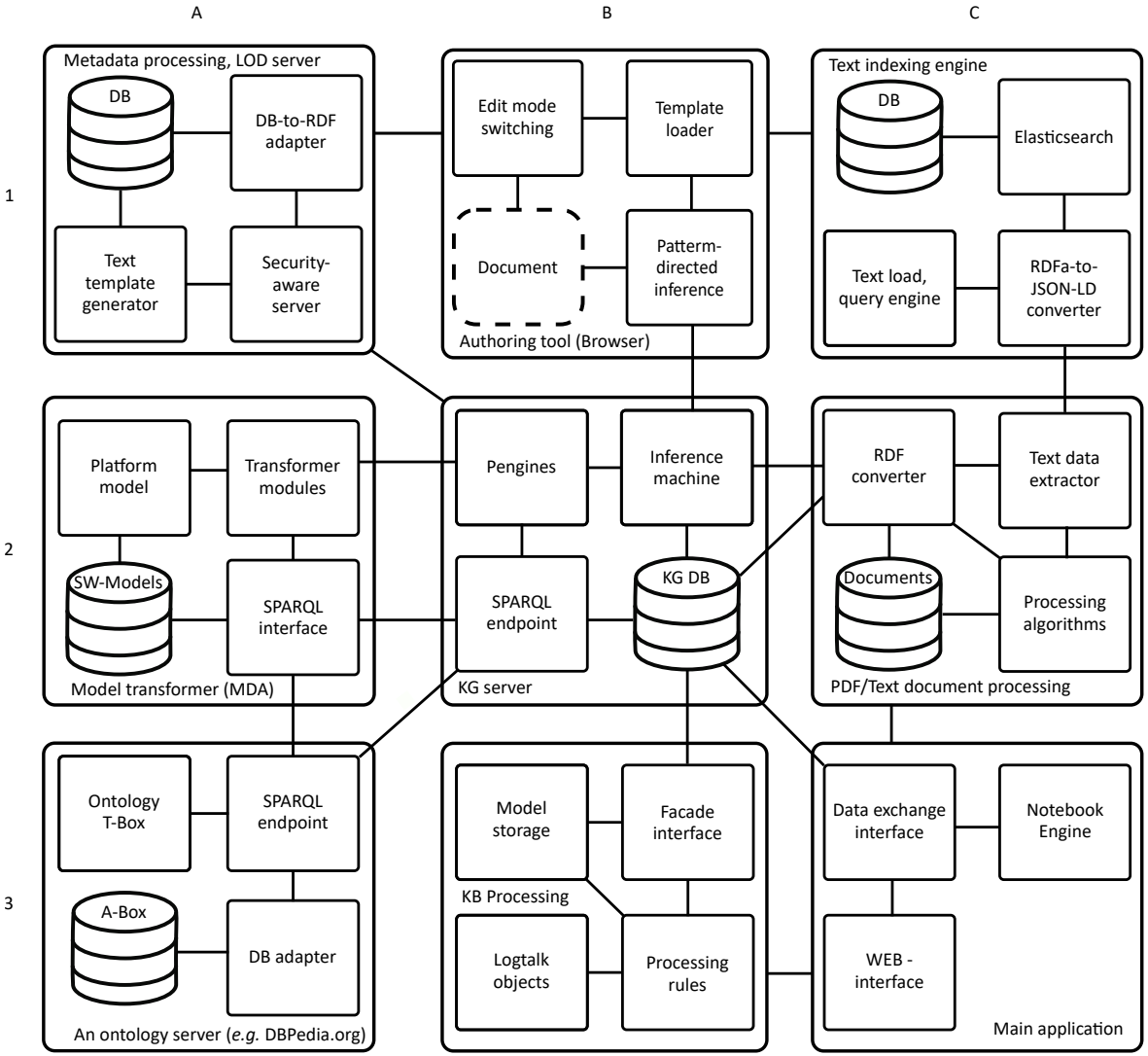
The problem set is as follows:

- CPs published on ISU website for the active students' groups
- Educational, industrial standards on the government site (some of them just image scans)
- Document templates produced by institute management staff
- Data of the developed applications

At the first stage, we deal with information acquisition and analysis to reach a reasonable level of formalization. This results to the following general requirements:

1. Allow loose coupling between application (agents) and independent development
2. Store data with metadata to form a standardized level of application interaction
3. Respect users' day-to-day way of task solution (principle of the least surprise)
4. Use perspective information technologies

Architecture of the infrastructure



Abbreviations

T-Module is Transformation module
MDA is Model-Driven Architecture
T-Box is Terminological Box
A-Box is Instance Box
KG DB is Knowledge Graph Database

Semantic web technologies & Knowledge graphs

Semantic Web (WEB 3.0) is characterized with

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

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

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

CPs PDF analysis

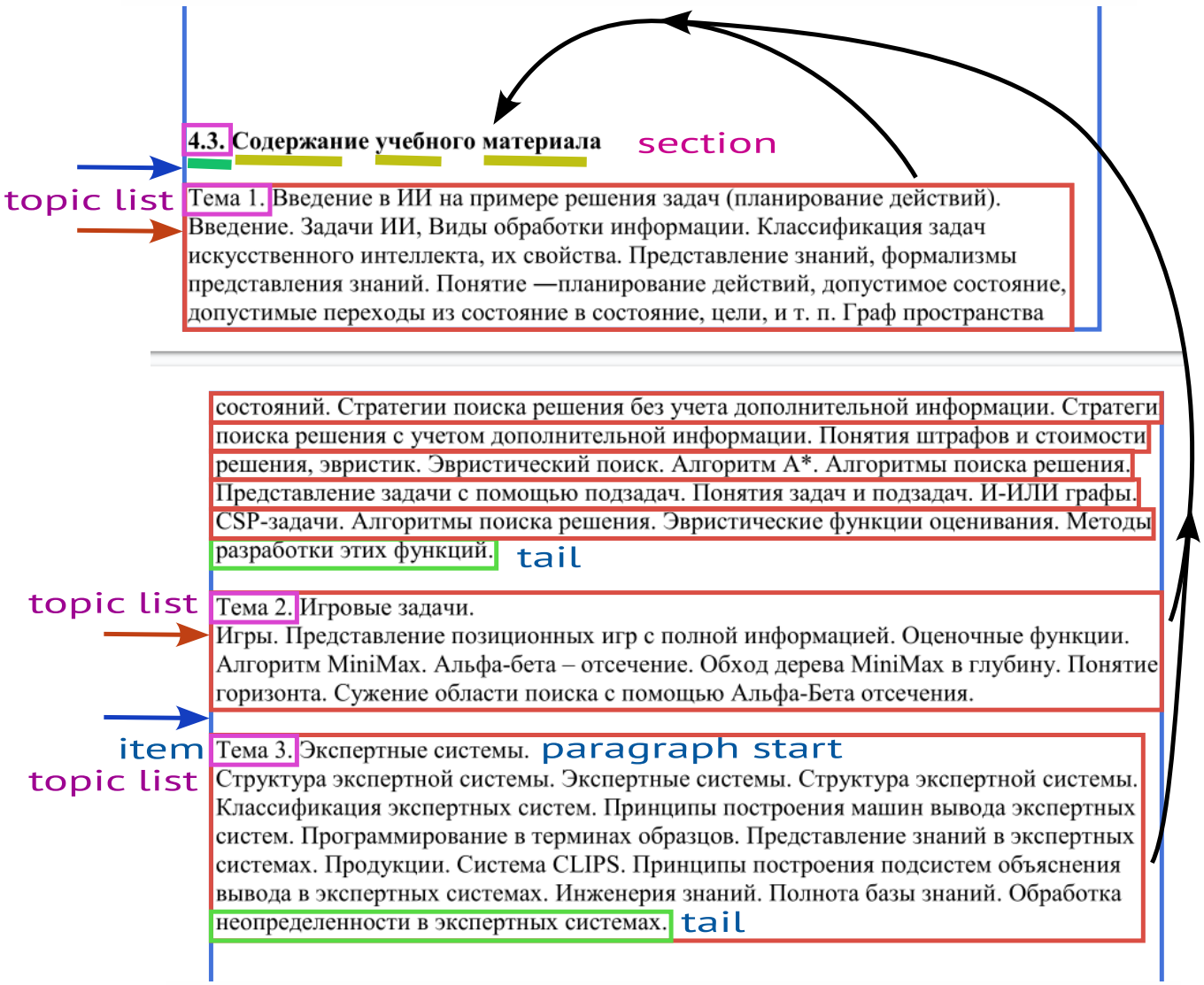
Meaningful information is the title and the code of the course, list of topics, distribution of study units (academic hours) between lectures, practice, seminars, personal work of student, list of questions for knowledge assessment, tests, *etc.*

1. Convert PDF to XML by `Poppler`'s `pdftohtml -s -c -xml`
2. XML is converted to in-object database of ordered `elements/5`
3. Text bounding box by non-empty runs, excluding page numbers
4. Line, run, font, page features assessment (indent, tail, numbers, ...)
5. Join runs in lines, lines in paragraphs, removing page breaks
6. Recognition of sectioning, association of paragraph to sections
7. Join and fold lists, assuming bullet lists have deeper level
8. Data acquisition respecting sectioning, store them in a KG and an HTML file

Steps 4 and 5 are repeated until no joins were performed.

Recognized KG data are filled in a LuaLaTeX template using an MVC framework.

Structure recognition



CPs recognition scenario for IMIT, ISU

```
:- object(CP_recognizer(_XML_, _HTML_File_Name_, _Document_IRI_),
        extends(as_db(_XML_)),  % A parametric object
        imports([CP_fonts, text_attrib, degraded, text_features,
                 CP_merge, text_CP_sections, gather_items,
                 CP_page_one, text_CP_fields, grouping,
                 htmlize])).    % Importing categories
   % . . . . . . . . . . . . . .
   % Configuration parameters of the recognizer
   deviation(attributes, [10, 50]).    % tolerances
   deviation(paragraph, [50, 10]).     %
   deviation(parindent, [28]).         % 1 cm = 28 pt
   deviation(itemtextminlength, [10]). % The length of a "minimal item text".
   :- public(process/0).               % public predicate
   :- info(process/0, [ comment is 'Run all rules in an order' ]).
   process :-
      ::process_CP_fonts, !, ::process_attrs, !,
      ::process_degraded, !,       ::process_runs_merge, !,
      ::process_features, !,       ::process_merge, !,
      ::process_features, !,       ::process_merge, !,
      ::process_first_page, !,     ::process_CP_sections, !,
      ::process_item_gathering, !, ::process_merge, !,
      ::process_CP_fields, !,::process_grouping(ul), !,
      ::process_merge, !,          ::htmlize(_HTML_File_Name_, _Document_IRI_),
      !.
:- end_object.
```

Extending line-joining category

```
:- category(CP_merge, extends(text_merge)).
    :- protected(lines_mergable/2).
    lines_mergable(A, B) :-  % Try to use default rules
        ^^lines_mergable(A, B), !. % Parent category predicate call
    lines_mergable(element(_, _, text, _, S1), element(_, _, text, _, S2)) :-
        ::gettext(S1, T1),     ::unterminated_sentence(T1),
        ::gettext(S2, T2),     ::cannot_start_sentence(T2), !.
    lines_mergable(element(_, _, text, A, _),  element(_, _, text, _, S2)) :-
        ::list_item(A),        ::gettext(S2, T2),
        ::cannot_start_sentence(T2), !.
    % . . . . . .
    :- protected(unterminated_sentence/1).
    unterminated_sentence(T) :-
        re_match("[-/+=—]\s*$", T, []).
    unterminated_sentence(T) :-
        string_lower(T, L),   re_match("url\s*:\s*$", L, []).
    % . . . . . .
    :- protected(cannot_start_sentence/1).
    cannot_start_sentence(T) :-
        re_matchsub("^\s*[a-я]+\s*([):]?)", T, Dict, []),
        get_dict(1, Dict, ""), !.
    % . . . . . .
:- end_category.
```

Result

4.3. Содержание учебного материала

Тема 1. Введение в ИИ на примере решения задач (планирование действий). Введение. Задачи ИИ, Виды обработки информации. Классификация задач искусственного интеллекта, их свойства. Представление знаний, формализмы представления знаний. Понятие —планирование действий, допустимое состояние, допустимые переходы из состояние в состояние, цели, и т. п. Граф пространства состояний. Стратегии поиска решения без учета дополнительной информации. Стратегии поиска решения с учетом дополнительной информации. Понятия штрафов и стоимости решения, эвристик. Эвристический поиск. Алгоритм A*. Алгоритмы поиска решения. Представление задачи с помощью подзадач. Понятия задач и подзадач. И-ИЛИ графы. CSP-задачи. Алгоритмы поиска решения. Эвристические функции оценивания. Методы разработки этих функций.

Тема 2. Игровые задачи. Игры. Представление позиционных игр с полной информацией. Оценочные функции. Алгоритм MiniMax. Альфа-бета – отсечение. Обход дерева MiniMax в глубину. Понятие горизонта. Сужение области поиска с помощью Альфа-Бета отсечения.

Тема 3. Экспертные системы. Структура экспертной системы. Экспертные системы. Структура экспертной системы. Классификация экспертных систем. Принципы построения машин вывода экспертных систем. Программирование в терминах образцов. Представление знаний в экспертных системах. Продукции. Система CLIPS. Принципы построения подсистем объяснения вывода в экспертных системах. Инженерия знаний. Полнота базы знаний. Обработка неопределенности в экспертных системах.

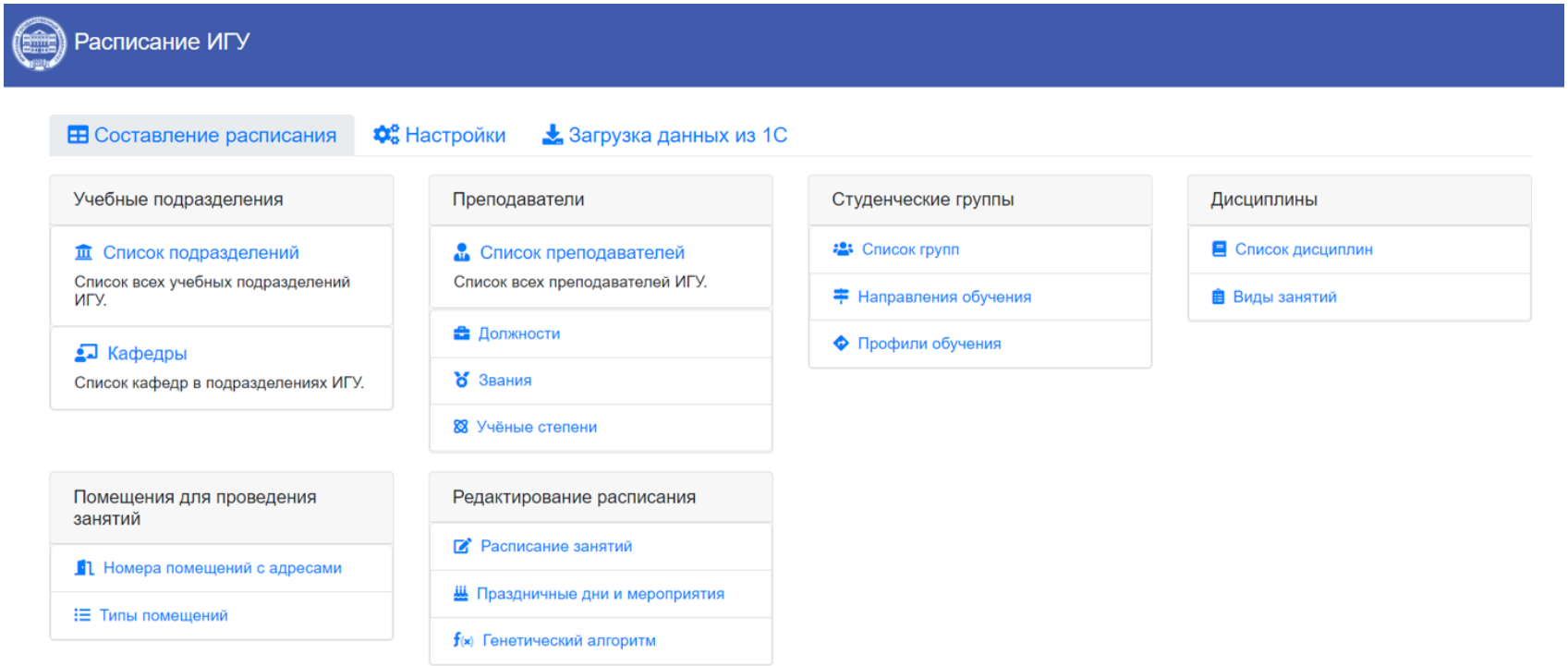
For the **title** and **topics**, structures of KG graph are created, as well as **relations between them**, accounting contexts.

“ISU Schedule” Functions

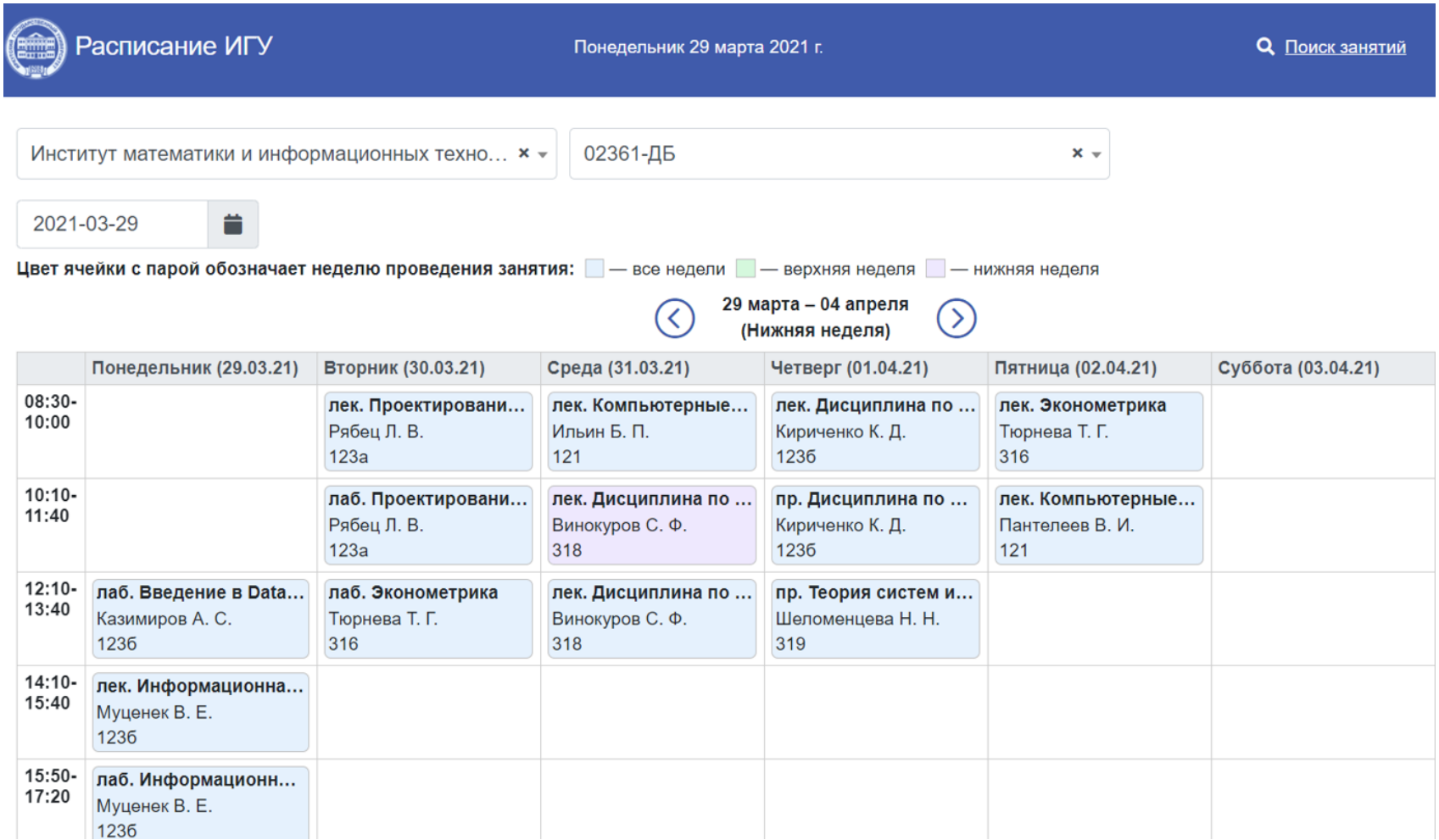
“ISU Schedule” is a web-application has the following functions:

- accounting main features of ISU institutions, such as:
  - different timetables
  - periodicity of classes
  - time spending for classes variety
- editing of the education timetables for mural and extramural groups
- printing schedules for an auditorium, course, faculty member, and student group
- register a user
- input holiday dates
- load data from “1C:University”
- schedule compilation with a genetic algorithm

Administration panel



A schedule view



	Понедельник (29.03.21)	Вторник (30.03.21)	Среда (31.03.21)	Четверг (01.04.21)	Пятница (02.04.21)	Суббота (03.04.21)
08:30-10:00		**лек. Проектировани…** Рябец Л. В. 123а	**лек. Компьютерные…** Ильин Б. П. 121	**лек. Дисциплина по …** Кириченко К. Д. 123б	**лек. Эконометрика** Тюрнева Т. Г. 316	
10:10-11:40		**лаб. Проектировани…** Рябец Л. В. 123а	**лек. Дисциплина по …** Винокуров С. Ф. 318	**пр. Дисциплина по …** Кириченко К. Д. 123б	**лек. Компьютерные…** Пантелеев В. И. 121	
12:10-13:40	**лаб. Введение в Data…** Казимиров А. С. 123б	**лаб. Эконометрика** Тюрнева Т. Г. 316	**лек. Дисциплина по …** Винокуров С. Ф. 318	**пр. Теория систем и…** Шеломенцева Н. Н. 319		
14:10-15:40	**лек. Информационна…** Муценек В. Е. 123б					
15:50-17:20	**лаб. Информационн…** Муценек В. Е. 123б					

Main program features

Editing mode functions

1. Adding classes for, a date, a time period, define periodicity
2. Update class data:
   - shifting a class to another day of week or a time period
   - changing the auditorium
   - assigning other teacher
3. Removing or cancellation of a class

Genetic algorithm constructs a schedule for an institute, accounting

- teacher day load
- the capacity of auditoriums
- disabling gaps for student groups and faculty

Program performance

For 30 student groups of IMIT, ISU

- An acceptable solution is obtained for 5000 iterations (40 minutes)

A program execution result

02342-ДБ

	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота
08:30-10:00						
10:10-11:40			**Компьютерные издательск…** Ильин Борис Петрович 113-3	**Технологии разработки про…** Чугунов Андрей Александрович 113-2		
12:10-13:40	**Введение в Data Mining** Казимиров Алексей Сергеевич 318	**Компьютерные издательск…** Ильин Борис Петрович 113-1	**Дисциплина по выбору "ра…** Кедрин Виктор Сергеевич 122	**Математическое моделиро…** Шеломенцева Наталья Николаевна 122	**Введение в Data Mining** Казимиров Алексей Сергеевич 113-2	
14:10-15:40	**Функциональное программ…** Хмельнов Алексей Евгеньевич 113-3	**Эконометрика** Тюрнева Татьяна Геннадьевна 201	**Эконометрика** Тюрнева Татьяна Геннадьевна 201	**Функциональное программ…** Хмельнов Алексей Евгеньевич 113-3	**Технологии разработки про…** Чугунов Андрей Александрович 123а	
15:50-17:20	**Информационная безопасн…** Муценек Витус Евгеньевич 113-3	**Информационная безопасн…** Муценек Витус Евгеньевич ЦНИТ 113-2	**Математическое моделиро…** Сорокин Степан Павлович 203б			
17:30-19:00		**Дисциплина по выбору "ра…** Кедрин Виктор Сергеевич 113-1				
19:10-20:40						

Conclusion

This is a progress report of R&D of a Knowledge Graph based architecture and infrastructure for automation of creative activity of a faculty. The following results were obtained:

1. Raw domain analysis, including, problem space
2. Existing software and data source accounting
3. Realized MVP-like utilities providing solutions of new problems
   - Knowledge Graph (KG) component infrastructure is being organized
   - Analyzing PDF-exported versions of CPs by a Logtalk knowledge-based system
   - Collecting data from the CPs documents and store in KG
   - Implementing verification software for parts of CP
   - Document authoring tools are being implemented using generative approaches
   - Techniques of Linked Open Data and standard vocabularies' usage is being formalized

Thanks for Your Attention!