Web-GIS viewer for active faults data represented as a knowledge graph

Evgeny A. Cherkashin[1,3,4], Oksana V. Lunina[2], Leonid O. Demyanov[3] and
Alexander V. Tsygankov[4]

[1]*Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences, 134 Lermontov St, Irkutsk, 664033, Russian Federation*

[2]*Institute of the Earth's Crust of Siberian Branch of Russian Academy of Sciences, 128 Lermontov St, Irkutsk, 664033, Russian Federation*

[3]*Institute for Mathematics and Information Technologies, Irkutsk State University, 20 Gagarina Bulv, Irkutsk, 664003, Russian Federation*

[4]*Institute for Information Technologies and Data Analysis, National Research Irkutsk State Technical University, 83 Lermontov St, Irkutsk, 664074, Russian Federation*

Abstract

A problem of flexible geographical data representation and Web-based visualization is considered. The data stored in a knowledge graph as ontologies (vocabularies) in accordance to W3C standards. For viewing data, a web geographical information system (GIS) application is realized, which renders map interpreting SPARQL queries to Sematic Web server storing the knowledge graph. The technologies used for designing are based on contemporary Web 3.0, allowing one to implement Linked Open Data (LOD) compliance for GIS information publishing and integration.

Keywords

geographical information system, knowledge graph, semantic web, storing flexible data, one-page web application

1. Introduction

Contemporary Web technologies development is aimed at more tight data integration: standardization of data publishing formats, formal data and metadata representation, unifying interpretation contexts, referring to external entities. Geospatially related objects are frequently published in the requirements as well.

Geospatial data are of primary interest of people as many human activities are attached to object located on the Earth's surface. There are a lot of on-line services helping users to navigate city streets, figure out places corresponding the required conditions (shopping centers, parking places, *etc.*), observation of territories to familiarize themselves. The objects of interest as any data objects are described with attributes of various kinds, like working hours of firms, their home site URLs, nearby bus stations. At present, these software products trend to allow

integration with their data via open formats and publishing principles, *e.g.*, Linked Open Data [1].

The fault data [2] are chosen as subject for representation and publishing in this investigation. In the geology, researchers accumulate data obtained after event observations, *e.g.*, earthquakes, landslides, by analyzing remote sensing data and results of field works. The obtained data are processed and interpreted, resulting in setting new attributes to a fault or refinement of their values. According to the techniques of geological research, additional information are associated with attributes, clarifying their values. Such clarification comprises precision characteristics, measurement conditions, reliability assessment, and paper references, where fault data were published.

GIS[1] represents spatial data in semantic defined layers. For each object of a layer, one can associate a set of attribute values of auxiliary data. The set of the attributes are the same for each object of the layer, regardless of whether the attribute value is defined for an object or not. Empty values are represented as "`null`"s. In the case of geological exploration, when a lot of attributes are undefined, this approach leads to sparse filled tables. This, in turn, requires data modification and analysis algorithms to utilize additional data checking stages when using standard relation operations (SELECT, UPDATE, DELETE).

Another question is attribute names definition expressing semantics of metadata. To define a precision of a value, one could construct an attribute name of "`<name>_prec`" structure, where `<name>` is its value attribute identifier. Other types of metadata add more suffixes, as well as relations between suffixes and values are not defined anywhere in the database. The formal definition is to be described in documentation or defined as processing algorithms. Thus, the implied semantics is either informally defined or obfuscated and practically is not alienated. Adding new attributes requires the user to devise new synthetic names.

Web publication applications, as information systems, are to implement filtering functions, differentiating the value attributes and their metadata. Screen widgets label names either defined in application configuration or figured out from the attribute names. Lack of ontological (vocabulary) formal domain definition forces developer to spend more efforts for the user interface implementation.

Since 2001, Semantic Web technologies have evolved in a substantial set of instruments for data storing, publishing, and software integration, allowing system designers and programmers, among standard means, to pass data between systems via published documents and application user interfaces, *i.e.*, extending their set of functions. Vocabularies and data instances are stored in graph databases, which provide SPARQL and other endpoints on top of HTTP protocols, providing services for data access and modification similar to relational database servers. The generalized problem statements and approaches to their solutions in the field of Semantic Web are the reasons of its constant development.

Knowledge graphs (KG) [3] are techniques of Semantic Web usage aimed at representation of data in a general flexible way allowing so-called "natural" evolving of domain image, including representation of incomplete knowledge. This evolution corresponds to a scientific research process, where data is permanently accumulated and analyzed. KG technologies provide distributed storage, federated query based access and modification, means for metadata definition,

---

[1]Abbreviation of Geographical Information System

```
!table                                          Width_damage_zone_Q Char (2) ;
!version 300                                    Slip_sense Char (30) Index 10 ;
!charset WindowsCyrillic                        Slip_sense_Q Char (2) ;
                                                Slip_sense_Index Decimal (2, 0) ;
Definition Table                                Total_Cenozoic_lateral_slip_m Char (20) ;
  Type NATIVE Charset "WindowsCyrillic"         Total_Cenozoic_lateral_slip_Q Char (2) ;
  Fields 72                                     Total_Cenozoic_vertical_slip_m Char (20) ;
    ID Char (15) Index 1 ;                      Total_Cenozoic_vertical_slip_Q Char (2) ;
    Name Char (40) Index 2 ;                    . . . . . . . . . . . . . . . . . . . .
    Location Char (250) Index 3 ;               Potential_Ms_max Float ;
    Strike Char (10) Index 4 ;                  Potential_Ms_max_Q Char (2) ;
    Strike_Q Char (2) ;                         Potential_Mw_max Float ;
    Dip_azimuth Char (10) Index 5 ;             Potential_Mw_max_Q Char (2) ;
    Dip_azimuth_Q Char (2) ;                    Elapsed_time_years Char (10) ;
    Dip_angle Char (10) Index 6 ;               Elapsed_time_Q Char (2) ;
    Dip_angle_Q Char (2) ;                      Associated_CSS Char (25) ;
    Length_km Char (10) Index 7 ;               Associated_IGGSS Char (50) ;
    Length_Q Char (2) ;                         Seismic_activity_of_fault Char (3) ;
    Depth_km Char (10) Index 8 ;                Compiler Char (50) ;
    Depth_Q Char (2) ;                          Date Char (10) ;
    Width_damage_zone_km Char (10) Index 9 ;
```

Figure 1: A part of the structure of the original database

and formalized verification of its content.

The aim of this research and development is to represent existing tabular and spatial data from [2, 4] as a knowledge graph with implementing a viewer, assessing "working efficiency" of a programmer, state further development perspectives, ranging them by priorities.

2. Data conversion

The original database table structure is shown in Figure 1. This structure contains a number of fields, namely, "ID" defining fault identifier, the fault name as a geographical entity, various characteristics with corresponding clarifications, seismic activity, name of compiler researcher and date of refinement. Data is stored in DBF format, the record number relates the database record with spatial object of fault layer. Field names in DBF file cannot be longer than ten characters, should be capitalized, number of fields cannot be more than 255.

Identifiers of the attributes with common prefixes define one value with a clarification, *e.g.*, "Depth" and "Length" of a fault are measured in kilometers, their values are clarified with "quality" attribute having suffix "_Q" at the end.

Table content slice of the fault database is shown in Figure 2. It is sparse filled: many attributes are nulls. The field "Geomorphol…" and some others are filled with values of their predefined sets. The consistency are controlled algorithmically with QGIS [5] extension modules. This flat structure and representation format are intended to be easily accessed and realized with standard relational database tools, but general violations of standard normal forms forces the developer to implement subroutines controlling a record content (semantics) in addition to the

	Total_Ce_1	Total_Ce_2	Total_Ce_3	Reliabilit	Geomorphol	Geomorph_1	Geophysica	Geophysi_1	Engineerin	Engineer_1	Gydrogeolo	Gydrogeo_1	Meteorolog	Meteorol_1	Structur
1	ЛС	NULL	ЛС	1	Серия линеам...	3	NULL	0	При Култукско...	0	После землетр...	2	NULL	0	Взбросовое
2	ЛС	300	ЛС	1	Линеамент на ...	3	NULL	0	NULL	0	NULL	0	NULL	0	NULL
3	ЛС	450	ЛС	1	Линеамент на ...	3	NULL	0	Сейсмогравит...	3	NULL	0	NULL	0	NULL
4	ЛС	700	ЛС	1	Линеамент на ...	3	NULL	0	NULL	0	NULL	0	NULL	0	NULL
5	ЛС	2000	ЛС	1	Уступ на топоо...	3	Сегмент разло...	1	NULL	0	NULL	0	NULL	0	Многочисл
6	ЛС	2000	ЛС	1	Уступ на топоо...	3	Выделяется в г...	1	Обвалы объем...	3	Восходящий в...	2	NULL	0	Крупный ск
7	ЛС	2000	ЛС	1	Уступ на топоо...	3	NULL	0	NULL	0	NULL	0	NULL	0	NULL
8	ЛС	2000	ЛС	1	Уступ на топоо...	3	NULL	0	NULL	0	NULL	0	NULL	0	NULL
9	ЛС	2000	ЛС	1	Уступ на топоо...	3	NULL	0	NULL	0	NULL	0	NULL	0	NULL
10	ЛС	2000	ЛС	1	Уступ на топоо...	3	NULL	0	NULL	0	NULL	0	NULL	0	NULL
11	NULL	NULL	NULL	1	Уступ и линеа...	3	NULL	0	NULL	0	NULL	0	NULL	0	Зоны трещи
12	NULL	NULL	NULL	1	Линеамент на ...	1	NULL	0	NULL	0	NULL	0	NULL	0	NULL
13	NULL	NULL	NULL	1	Линеамент на ...	1	NULL	0	NULL	0	NULL	0	NULL	0	NULL
14	NULL	NULL	NULL	1	Линеамент на ...	3	Крупные сейс...	3	NULL	0	NULL	0	NULL	0	NULL
15	NULL	NULL	NULL	1	Линеамент на ...	3	Изгибы кровл...	1	NULL	0	Восходящий и...	2	NULL	0	Сейсмиты
16	NULL	NULL	NULL	1	линеамент на ...	3	Обвалы объем...	0	NULL	3	NULL	0	NULL	0	NULL
17	NULL	NULL	NULL	1	Серия линеам...	1	NULL	0	NULL	0	NULL	0	NULL	0	NULL
18	NULL	25	ЛС	1	Линеамент на ...	3	Крупные сейс...	0	NULL	3	NULL	0	NULL	0	NULL
19	NULL	NULL	NULL	1	Линеамент на ...	1	NULL	0	NULL	0	NULL	0	NULL	0	NULL
20	NULL	NULL	NULL	1	Серия линеам...	1	NULL	0	Горячий источ...	0	NULL	2	NULL	0	NULL
21	NULL	NULL	NULL	1	Изгибы рек на ...	1	NULL	0	NULL	0	NULL	0	NULL	0	NULL
22	NULL	NULL	NULL	1	Серия линеам...	1	NULL	0	NULL	0	NULL	0	NULL	0	NULL
23	NULL	NULL	NULL	1	Линеамент на ...	1	NULL	0	NULL	0	NULL	0	NULL	0	NULL
24	NULL	NULL	NULL	1	Отдельные сла...	1	NULL	0	NULL	0	NULL	0	NULL	0	NULL
25	NULL	NULL	NULL	1	Серия мелких ...	1	NULL	0	NULL	0	NULL	0	NULL	0	NULL

Figure 2: A table content slice

UPDATE DML[2] command.

Conversion of the table to a KG started with defining T-Box[3] comprising the class of fault and its subclasses. Subclasses correspond to various kinds of faults, *e.g.*, normal, reverse, strike-slip, oblique. As a differentiating property, "slip_sense" attribute was used. All the fault records have been converted into triplets and assigned a class. The resulting set of triplets formed A-Box[4], a KG of fault instances. Non-null attributes of faults were converted into a literal relation, except those, which values were restricted to a finite set. These attribute values presented as references to a descriptive constant added to T-Box. These conversions were implemented as Python program loading DBF-files and generation OWL2 XML ones. After the conversion, the obtained OWLs of T-Box and A-Box were visually checked with Protégé, saved into Turtle (ttl) format for the further use.

The results of conversions were manually loaded into GraphDB and Jena servers. For each part (T- and A-Box) a global namespace (aft, af) were allocated: http://irnok.net/ontologies/ActiveFaultTerms# and .../ActiveFault#. In the software, for each KG one must set up an individual endpoint providing access. GraphDB user interface were used to check the correctness of class-instance relations with executing SPARQL-queries. The peculiarity of GraphDB usage is the necessity to "activate" each KG endpoint via user interface after loading their KG contents. After conversion and set up of the endpoint, they will be available at port 7200, URL will be formed out of server address and endpoint name. An example of a converted item is shown here

---

[2]Abbreviation of Data Manipulation Language.

[3]Abbreviation of Terminological Box, a set of basis domain terms and their relationships.

[4]Denotation of Instance Box, a set of the instances.

```
###  http://irnok.net/ontologies/ActiveFaults#RUAF_996          aft:unit aft:Kilometer ] ;
af:RUAF_996 rdf:type owl:NamedIndividual ,             aft:Location "At the edge of Barguzin depression and \
      aft:Fault ,          # Classification                Ikatsky ridge"^^xsd:string ;
      aft:NormalSlCB ,                                  aft:Meteorological_grade [ aft:value "0"^^xsd:int ] ;
      aft:PlioceneFault ;                               aft:Paleoseismological_grade [ aft:value "0"^^xsd:int ] ;
  aft:ID "RUAF_996"^^xsd:string ;                       aft:Potential_Ms_max [ aft:value "0.0"^^xsd:float ] ;
  aft:Activity_degree aft:Light ;                       aft:Potential_Mw_max [ aft:value "0.0"^^xsd:float ] ;
  aft:Averaged_slip_rate [ aft:value "0.0"^^xsd:float ; aft:Reliability_class [ aft:value "1.0"^^xsd:float ] ;
      aft:unit aft:Millimeter ] ;                       aft:Seismic_activity_of_fault [ aft:value "false"^^xsd:boolean ] ;
  aft:Compiler "Lunina O.V."^^xsd:string ;              aft:Seismological_grade [ aft:value "0"^^xsd:int ] ;
  aft:Date "29.05.2011"^^xsd:string ;                   aft:Slip_rate_grade [ aft:value "0"^^xsd:int ] ;
  aft:Dip_azimuth [ aft:value "310"^^xsd:string ;       aft:Slip_sense [ aft:value aft:Discard ;
      aft:quality aft:LC ] ;                                aft:index "1.0"^^xsd:float ;
  aft:Engineering_geological_grade [ aft:value "0"^^xsd:int ] ; aft:quality aft:LC ] ;
  aft:Geomorphological_features [                       aft:Strike [ aft:value "40"^^xsd:string ;
      aft:value "Topographic ledge: 1 point"^^xsd:string ;  aft:quality aft:LC ] ;
      aft:grade "1"^^xsd:int ] ;                        aft:Structural_geological_grade [ aft: value "0"^^xsd:int ] ;
  aft:Geophysical_grade [ aft:value "0"^^xsd:int ] ;    aft:Total_activity_grade [ aft:value "1.0"^^xsd:float ] ;
  aft:Gydrogeological_grade [ aft:value "0"^^xsd:int ] ; aft:Total_max_slip_per_event [aft:value "0.0"^^xsd:float ;
  aft:Last_activation_age [ aft:value aft:Pliocene ;        aft:unit aft:Meter ] ;
      aft:ageindex "4.0"^^xsd:float ] ;                 aft:Vertical_max_slip_per_event [ aft:value "0.0"^^xsd:float ;
  aft:Lateral_max_slip_per_event [ aft:value "0.0"^^xsd:float; aft:unit aft:Meter ] ;
      aft:unit aft:Meter ] ;                            aft:Width_damage_zone [ aft:value "1.21"^^xsd:string ;
  aft:Length [ aft:value "12.09"^^xsd:string ;              aft:quality aft:AC ;
      aft:quality aft:LC ;                                  aft:unit aft:Kilometer ] .
```

3. Viewer implementation

Viewer application is a Web-2.0 browser widget composed out of React components. The application maintains its state with Redux. Properly used Redux allows one to control viewer widgets content and structures solely by defining state change functions. The state transitions happened whenever a set of visualized faults has changed.

Rendering maps is based on React wrapper of `leaflet.js` library. The library allows one to draw interactive graphics objects (polygons) on a topological basis. Leaflet supports various sources of the bases, such as openstrrtmpa.org, which is default one. Fault shapes were converted from KML format to leaflet interpretable JSON and are being loaded by application on demand from web server.

By default, the viewer shows all faults from KG, the result of a SPARQL query requesting all objects of a class `aft:Fault`[5], *i.e.* all the instances[6]. There is a form in the user interface for setting filtering conditions. They change the default SPARQL query, adding restrictions. For the debugging and developing, one version of the viewer contains a form for free-form SPARQL query definition, which execution results also showed as a fault map.

Viewing fault attributes is implemented as a form document improved with RDFa markup. This markup originates from results of SPARQL query selecting all the predicate relation for the chosen fault. In Figure 3, viewer window is shown. The image consists of a topological basis loaded from server `openstreetmap.org`, faults drawn over the basis, filter form at the left-hand side, expanding form of free-text SPARQL query, popup with fault name pointing to the fault shape, and a form showing known attributes of the fault. As a subject of interest, Irkut-Ushakovsky fault extended through Irkutsk city is chosen.

---

[5]Abbreviated namespace synonym of http://irnok.net/ontologies/ActiveFaultTerms#Fault.

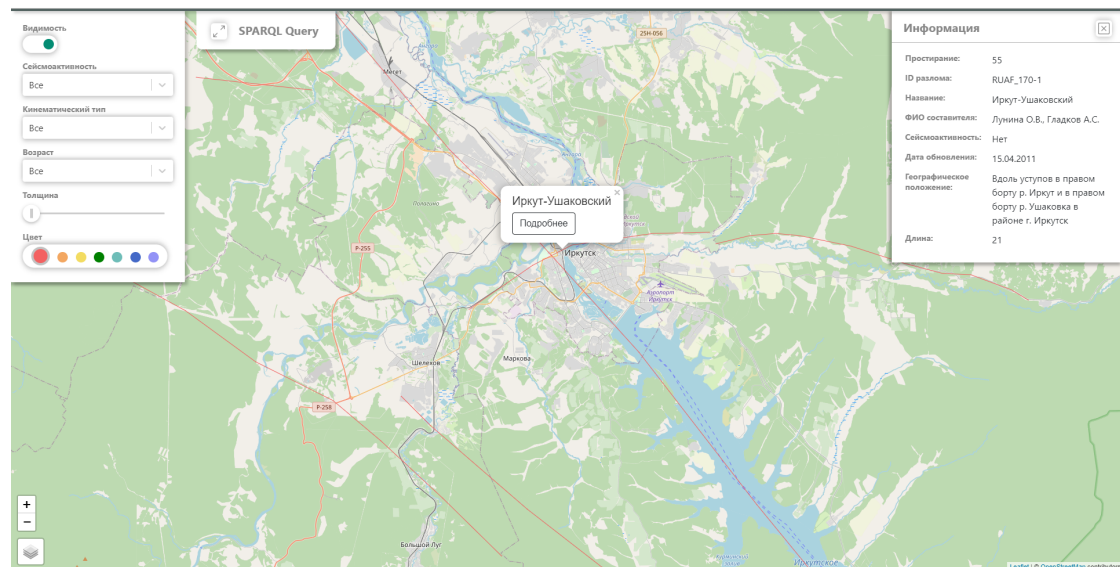[6]All faults are instances of `aft:Fault`.

Figure 3: Viewer window showing description of a fault

4. Related Works

The activity related to the field of our investigation are observed since beginning of 2010-th. At that time probably a lot of vocabularies were standardized, their representation data formats become widely used, as well as various XML processing techniques and tools were developed, *e.g.*, XPath, XQuery, XSL[7]. SPARQL standards were acknowledged (2008, v. 1.0, 2013, v. 1.1). Thus, in early 2010-ths Semantic Web becomes not only academic field of interest, but also a set of technologies, oriented on solving practical problems. We would like to mention the following projects, representing the state of the art in the domain.

The project [6] was to represent OpenSteetMap (OSM) data as a KG, researchers

- Resemble the `DBPedia.org` project formalizing Wikipedia data but over the OSM database;
- Convert OSM data into RDF adhering LOD principles;
- Designed a vocabulary for object georeferencing;
- Related the objects to DBpedia, GeoNames, various icon sets;
- Developed a taxonomy of the objects on various levels (road → way (list of nodes));
- Stated the relations between nodes and means for defining complex objects;
- Implemented REST and SPARQL endpoints for actual data;
- Realized a live updates services from OSM change sets.

Thus, it was probably the first work no representation of spatial data as a KG.

GeolLink KG project [7] is aimed at representation various aspects of complex investigations of Earth's surface and bowels.

---

[7]Abbreviation of eXtensible Stylesheet Language, a technology for processing XML structures.

- KG includes diverse information as port calls made by oceanographic cruises, physical sample metadata, research project funding and staffing, and authorship of technical reports;
- Implements LOD (4 of 5 stars) and federated SPARQL integration between distributed parts of KG;
- Contains 45 millions RDF triples with vocabularies and geovisualization tools
- Describes interlinked expeditions (**R2**), oceanography (**BCO-DMO**), ocean floor microbiome (**IODP**), marine life papers (**MBLWHOI**), rock samples (**SESAR**), metadata of external research (**DataONE**), projects & conferences (**AGU-NSF**), sediment geochemistry (**NGDB**), Antarctica ice (**USAP**).

In the project, an update procedure (harvesting) is implemented to ensure the consistence of the KG w.r.t. the geo-base ontology (GBO).

The project [8] deals with developing a web GIS automatically publishing DBPedia data. The aim was to test Web-application tools capabilities in implementing GIS, which publishes celebrities' information living in a hometown/city. This is realized respecting LOD and Open Government Data principles. GIS module was realized on the topological basis and API v3 of Google. Celebrities' attribute data is loaded from DBPedia with SPARQL queries and rendered with a jQuery's Data table plug-in.

The project [9, 10] goal is to convert various existing GIS data into explicit knowledge, thus, forming a Spatial Data Infrastructure (SDI). The following requirements are to be met.

- Integrate existing geoportal data into a KB, including dynamic data;
- Geoportal data must be LOD, *e.g.*, HTML is enriched with RDFa;
- KG relation interpreters are implemented as knowledge based systems (named as expert systems), for example, "building near forest";
- Semantic enrichment of raw data to make it more usable/discoverable;
- Spatial properties of objects are figured out and processed with specialized GIS;
- Targeting to GeoSPARQL (sfIntersects, sfOverlaps, sfTouches, sfWithin, sfContains) language extension;
- Metadata inference from the data source properties;

The developed technologies are used to integrate public services data in Mazowieckie Voivodeship of Poland accounting European Union Open Government Initiative. For the user, the filtering queries are realized by interpreting a limited set of keywords.

5. Future activity plan

In our project, from the far perspective, we are to develop tools for modeling natural phenomena, *e.g.*, distribution of pollutant elements originating from faults and anthropogenic sources. And in this paper, we investigate Web 3.0 technologies for constructing nowadays WEB-GIS software.

Further development plan is as follows.

1. Improve the structure of the fault KG by means of adding various event data (earthquakes, landslides);
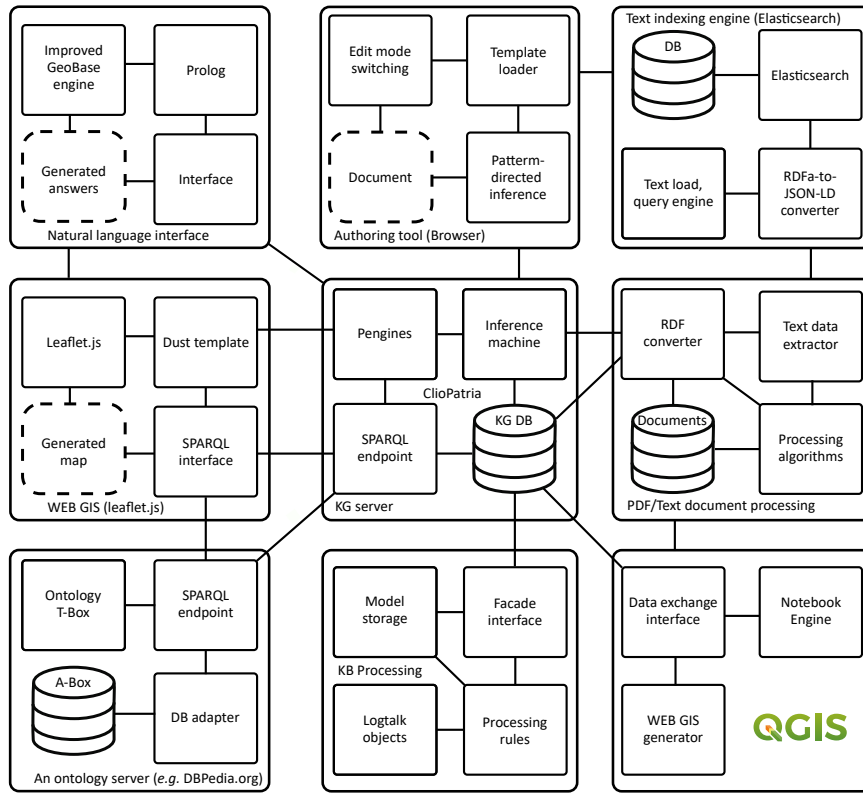
Figure 4: Target architecture of the environment

2. Add textual data and paper references for informal justification of KG content;
3. Provide a natural language interface for automation of complex filtering condition input;
4. Implement more on-demand interface elements loading, and function for choosing "similar" objects;
5. Implement editing of the KG and spatial data, *i.e.*, adopt the corresponding leaflet functionality;
6. Realize a bidirectional versioned data transfer between user's GIS and KG;
7. Attach existing World fault open data resources to the viewer;
8. Implement various analytical functionality for domain problem-solving;

The target infrastructure architecture is shown in Figure 4 [11]. The main bonding technologies of the environment are services represented as components interacting via HTTP, and knowledge based data processing implemented in SWI-Prolog and Logtalk logic programming languages. The final research result reports are partially generated by means of authoring tools, which are developed by our research group [12].

Conclusion

The proposed technology and software allows one to construct Web-GIS systems for research communities, as they support constant data accumulation, aggregation and analysis thanks to the properties of knowledge graph (KG) data storage and processing. The following properties of KG can be utilized for providing research environment

- Object data, relations, and metadata (vocabularies) are representable in KG;
- Processes of **node formation** and **improvement** of graph structure are done in parallel, *e.g.*, with SPARQL UPDATE and CREATE queries;
- Developers can **postpone** the formal definition of data schemata;
- KG is interpreted in three types of fundamental schemata:
    - **semantic**, aimed at representation of basis relations and type structures,
    - **validating**, *e.g.*, diligent formal definition in a KG can be verified w.r.t. sets of interpreting rules, and
    - **emergent** aimed at inferring more generalized structures from the current KG content and **reconstruction** the KG structure.

In the most of the domains, the existing data can be easily converted to a KG, loaded to a KG server and be accessed via SPARQL endpoints. A familiar tabular representation can be reconstructed with queries. Spending a reasonable time finding relevant to the domain vocabularies (ontologies) and adapting data conversion procedures to these vocabularies, one can obtain a common model description of a problem and shift data publishing and integration to a higher level.

In order to construct an environment for natural phenomena spatial modeling and master KG technologies, we created a GIS viewer for existing data converted in a KG [2, 4]. KG structures allowed us to improve structure of data originated from tables with forced violations imposed by a tabular representation of the scientific research data. The viewer interprets SPARQL queries results as a map. The nowadays Web 2.0 and 3.0 technologies allowed us to construct individual GIS application in a reasonable time, namely, two students constructed the application MVP in two months. This is thanks to the present levels of technologies and quality of used libraries.

Acknowledgments

References

[1] C. Bizer, T. Heath, T. Berners-Lee, Linked Data – The Story So Far. Int. J. Semantic Web Inf. Syst., 5 (2009), pp. 1–22. DOI:10.4018/jswis.2009081901

[2] O. V. Lunina. The digital map of the pliocene quaternary crustal faults in the southern east siberia and the adjacent northern Mongolia. Geodynamics & Tectonophysics. 7(3) (2016). pp. 407-434. DOI:10.5800/GT-2016-7-3-0215

[3] A. Hogan, E. Blomqvist, M. Cochez, C. D'Amato *et al.* Knowledge Graphs, 2020. https://arxiv.org/abs/2003.02320v5

[4] A. A. Gladkov, O. V. Lunina. Cartographic service "Activetectonics". http://activetectonics.ru/ (access date: 20-Sep-2021)

[5] M. Leidig, R. Teeuw. Free software: A review, in the context of disaster management. International Journal of Applied Earth Observation and Geoinformation, 42 (2015), pp. 49-56. DOI:10.1016/j.jag.2015.05.012.

[6] C. Stadler, J. Lehmann, K. Höffner, S. Auer. LinkedGeoData: A core for a web of spatial open data. Semantic Web 3 (2012) 333–354. DOI:10.3233/SW-2011-0052

[7] M. Cheatham, A. Krisnadhi, R. Amini, P. Hitzler, *et al.* The GeoLink knowledge graph, Big Earth Data, 2:2 (2018), pp. 131-143. DOI:10.1080/20964471.2018.1469291

[8] T. Abid, H. Zarzour. Integrating linked open data in geographical information system. Procs. of. International Conference on Information Technology for Organization Development. Oct 19-20, 2014, University of Tebessa, Tebessa, Algeria (2014).

[9] A. Iwaniak, I. Kaczmarek, M. Strzelecki, J. Lukowicz, P. Jankowski. Enriching and improving the quality of linked data with GIS. Open Geosciences, Vol. 8, 1 (2016) pp. 323-336. DOI:10.1515/geo-2016-0020

[10] A. Iwaniak, M. Leszczuk, M. Strzelecki, F. Harvey, I. Kaczmarek. A Novel Approach for Publishing Linked Open Geodata from National Registries with the Use of Semantically Annotated Context Dependent Web Pages. International Journal of Geo-Information. 6, 252 (2017). DOI:10.3390/ijgi6080252

[11] E. Cherkashin, A. Shigarov, V. Paramonov. Representation of MDA transformation with logical objects. International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Novosibirsk, Russia. (2019) 0913–0918 DOI:10.1109/SIBIRCON48586.2019.8958008

[12] E. Cherkashin, A. Shigarov, V. Paramonov, A. Mikhailov, Digital archives supporting document content inference, Procs. of 42-nd International Convention on Information and Communication Technology Electronics and Microelectronics (MIPRO), May, 20–24, 2019. pp. 1037-1042. DOI:https://doi.org/10.23919/MIPRO.2019.8757196

A. Online Resources

The sources for the viewer are being developed at Github https://github.com/De17eon/GRL.