Why use RDF/OWL rather than XML to represent and share global Legal Entity Identifiers (LEIs) and related LEI reference data

WHITE PAPER

ver. 1.0,
Publication date: 10.02.2016

Authors: Robert Trypuz, Dominik Kuziński
MakoLab S.A. Semantic Web Research and Development Team

EXECUTIVE SUMMARY

This White Paper describes some advantages of RDF/OWL representation of global LEI data in contrast to the current XML representation.

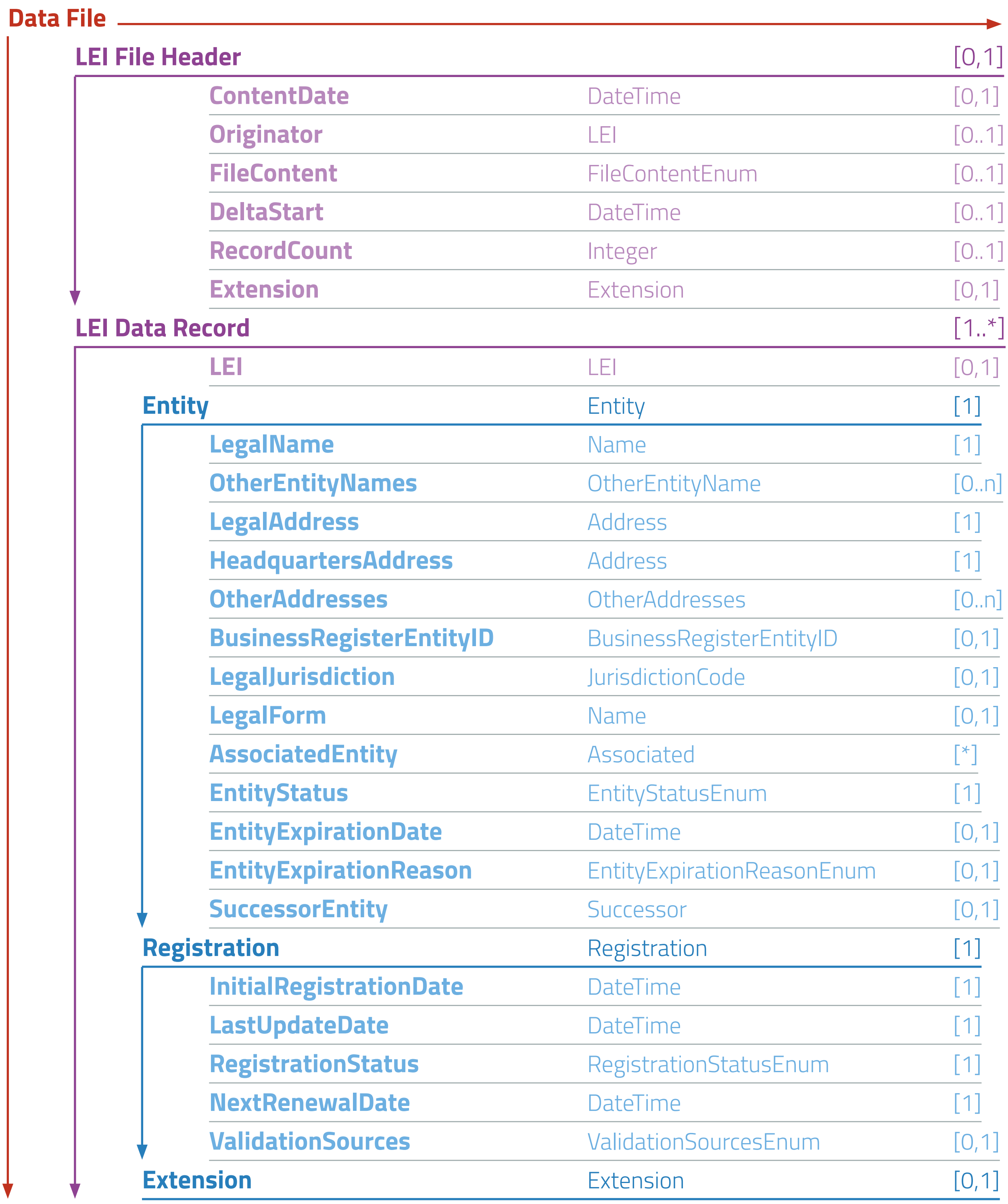
We propose a concrete RDF/OWL specification of LEI data named General Legal Entity Identifier Ontology (GLEIO) that is not only compliant with Common Data File Format but also allows for representation of changes of LEI and related LEI reference data over time.

We focus on benefits that can be gained by using semantic representation. Among them are: clear semantics of the LEI data, global persistent identification of the entities allowing for dereferencing URIs and retrieving information about the entities, flexibility of the representation allowing for easy extensibility (e.g., in the scope of Linked Open Data environment), possibility of using the content negotiation mechanism and finally, easy accessing and sharing of current and historical LEI data by the SPARQL endpoint.

The White Paper also presents MakoLab's web application that uses GLEIO to serve information about LEI and legal entities also taking into account their changes in time.

ABOUT COMMON DATA FILE FORMAT

The Common Data File Format is a standard by which a global Legal Entity Identifier (LEI) and legal entity reference data are reported by local operating units (LOUs). LOUs' reports are expressed in XML files defined by XML schema compliant with the Common Data File Format.



Data File		
LEI File Header		[0,1]
ContentDate	DateTime	[0,1]
Originator	LEI	[0..1]
FileContent	FileContentEnum	[0..1]
DeltaStart	DateTime	[0..1]
RecordCount	Integer	[0..1]
Extension	Extension	[0,1]
LEI Data Record		[1..*]
LEI	LEI	[0,1]
Entity	Entity	[1]
LegalName	Name	[1]
OtherEntityNames	OtherEntityName	[0..n]
LegalAddress	Address	[1]
HeadquartersAddress	Address	[1]
OtherAddresses	OtherAddresses	[0..n]
BusinessRegisterEntityID	BusinessRegisterEntityID	[0,1]
LegalJurisdiction	JurisdictionCode	[0,1]
LegalForm	Name	[0,1]
AssociatedEntity	Associated	[*]
EntityStatus	EntityStatusEnum	[1]
EntityExpirationDate	DateTime	[0,1]
EntityExpirationReason	EntityExpirationReasonEnum	[0,1]
SuccessorEntity	Successor	[0,1]
Registration	Registration	[1]
InitialRegistrationDate	DateTime	[1]
LastUpdateDate	DateTime	[1]
RegistrationStatus	RegistrationStatusEnum	[1]
NextRenewalDate	DateTime	[1]
ValidationSources	ValidationSourcesEnum	[0,1]
Extension	Extension	[0,1]

FIG 1. *https://www.gleif.org/lei-focus/what-is-an-lei/common-data-file-format*

XML schema for the Common Data File Format consists of (optional) LEI File Header and (mandatory) LEI Data Record (see figure 1 above). LEI File Header describes the source of data (e.g. by providing its content date). Each LEI Data Record provides information about a global LEI code and its registration. Here, one can also find information about a legal entity itself such as its global LEI, legal name, headquarters address, legal form, etc. XML schema also gives a space for extensions where additional data about a global LEI record (not defined in this standard) can be introduced.

DRAWBACKS AND LIMITATIONS OF XML REPRESENTATION

The role of the XML schema is to specify what constitutes a valid document. Thus XML schema compliant with the Common Data File Format is merely focused on syntax and structure of XML documents. It restricts the set of elements that can be used in the files with global LEI data. But it doesn't indicate how the documents should be interpreted. This leaves a room for misinterpretation while processing data.

Lack of semantics

The role of the XML schema is to specify what constitutes a valid document. Thus XML schema compliant with the Common Data File Format is merely focused on syntax and structure of XML documents. It restricts the set of elements that can be used in the files with global LEI data. But it doesn't indicate how the documents should be interpreted. This leaves a room for misinterpretation while processing data.

Thus, XML schema for the Common Data File Format cannot be understood by itself. To understand it one has to study LEI Data File Format 1.0 (http://www.leiroc.org/publications/gls/lou_20140620.pdf), i.e., a document where the Common Data File Format is described. The meaning of the element names in the document comes from the natural language description. E.g. element name "register" of type "BusinessRegisterEnum" is described there as follows: "A code that identifies the business register or other registration authority that supplied the value of EntityID." This description is confusing because type "BusinessRegisterEnum" suggests that we deal here with information about business registries but in the description we find that "other registration authority" is also allowed. Moreover in XML files (e.g. in 20160122-GLEIF-concatenated.xml) we do not really have codes for business registries (as the specification suggests). For instance a legal entity with LEI "391200VHUEDTABBEDF91" has business register "Handelsregister des Kanton Zug".

XML files are error prone. For instance, in the same XML file we find that a legal entity with LEI code "391200012TJB5Z15QN05" has business register "Handelsregister des Kantons Zug". Do the names "Handelsregister des Kanton Zug" and "Handelsregister des Kantons Zug" refer to the same register? It seems so, but we cannot be sure.
Processing the XML data is difficult when the data is a free (i.e., unrestricted) text. For instance a legal entity having global LEI code "391200014QLC3E6FR793" has a legal address where the city is named „Brunico / Bruneck (BZ)". Here, we have the original and translated name with code in the parentheses.
Cardinality in XML schema is also problematic. For instance cardinality „[1]" for Legal Address means that the element appears at most once in scope of the one LEI Data Record node. It is possible then that the same legal entity appears in two XML files, each time with exactly one legal address (so the XML restriction is satisfied), but the addresses are different. Is it intended? To prevent this situation it is necessary to define identity criteria for an address on the semantic level.
It is also worth noting that legal entities in the Common Data File Format have no identification. In the registration history of legal entity in LOU, we may find one or more general LEI code associated with it, so LEI code does not provide identity criterion for the legal entity.

Difficult extensibility

An XML document, defined by an XML schema, is not easily extensible. Adding new attributes to XML schema requires the agreement of all parties using the schema. The XML schema for Common Data File Format allows for free extensions, i.e. without any change in the schema. But adding new attributes to XML, without proper change in its schema, would result in parties involved not knowing what the data is about.

Lack of global identifiers

By global identifiers we mean here the unique resource identifiers (URIs) that are unique in the world. Unique identifiers can be added in XML documents, however, they are unique within the document and not globally. What can global identifiers contribute to the Common Data File Format?
By global identifiers and proper server setting (according to Linked Data principles), we can access information about LEI as we normally access webpages. The first step is to present content in a human readable way. GLEIF provides that possibility. For example, with the URL
https://www.gleif.org/lei/391200014QLC3E6FR793

we access the web page containing information about the global LEI with the code „391200014QLC3E6FR793". But this URL to the information about global LEI may not be persistent. Is it guaranteed that the LEI with the code „391200014QLC3E6FR793" will retain the URL „https://www.gleif.org/lei/391200014QLC3E6FR793" identification over the long term? A Persistent URL (PURL) concept may help here. PURL is an address on the World Wide Web that causes a redirection to another Web resource. It is particularly useful when a Web resource changes location (and hence URL) - a PURL pointing to the resource can be updated.

The second step is to allow for *content negotiation* which enables to serve different representations of the data at the same URI. This way, user agents can specify which response or model of representation they want to read (a human reader gets a website, an application gets structured data in e.g. JSON serialization).

Different URIs, even belonging to different conceptual structures, can be linked (by Linked Data properties). Benefits are obvious. We do not need to always start building our representation from scratch. If there is an external resource (e.g. ontology of ISO 3166 codes for countries) we can reuse it in our work.

Lack of inference

XML wasn't developed to be a subject of logical inference. Coherency and consistency checks are not possible or at least very difficult. It is of course possible to check whether the structure of an XML document is valid according to the predefined schema. However, many interesting constraints on data cannot be stated, e.g., LEI code length cannot be restricted inside the schema. GLEIF states that „original source files are not validated by GLEIF from a content perspective. They are only checked technically to verify compliance with the XML Schema of the common data file format".

In more advanced forms of representation like RDF/OWL, inference contributes to the meaning of elements (each inferred statement involving concepts contributes to their meaning). The more statements about something we have, the more we know about it.

Difficult access to LEI data by external applications

LEI data is stored on GLEIF's website in the compressed XML concatenated files.
Each XML file weighs around 500 MB. They are published every day with proper date (it is worth noting here that a current file is more likely heavier than the one from yesterday). Processing them by external applications is not easy. The files have to be first downloaded and then parsed. Comparative analysis of a few XML concatenated files is even more complicated.

SOLUTION: RDF/OWL MODEL OF THE COMMON DATA FILE FORMAT PUBLISHED AS LINKED OPEN DATA

WHY RDF/OWL IS A GOOD CHOICE

RDF/OWL solves all the challenges listed above.

Resource Description Framework (RDF) is a standard model for data interchange on the Web. Since 1999 RDF has been specified and recommended by W3C. It allows to make statements about web resources in the form of triples, i.e. subject–predicate–object expressions. The meaning of RDF data is specified by Web Ontology Language (OWL). OWL is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things. OWL is a computational logic-based language such that knowledge expressed in OWL can be processed by computer programs. OWL documents, known as ontologies, can be published in the World Wide Web and may refer to or be referred from other OWL ontologies.

Ontologies allow to classify entities and have enough expressive power to introduce constraints on the level of things. For example, we can formally express that „Each global LEI identifies exactly one legal entity" or that „an expiration reason can be specified only for those legal entities which have an expiration date". Such constraints indicate to the applications how global LEI data should be interpreted.

Semantic relations, for instance parthood, can help in linking collected information. For instance, Subdivisions and Countries can be linked by parthood relation (e.g. Nordrhein-Westfalen *is part of* Germany). As a result the data may be queried in a new way. For instance, having expressed in a semantic LEI data representation relations between Germany and its states e.g. by parthood, it is easy to issue a request in the SPARQL query language:

„Find a number of LEI codes for each German state."

Since RDF and ontologies link things (rather than describing the structure of documents), they are flexible enough to allow for extensions. New data based on terminology from external resources can be added to an RDF graph.

It is possible because RDF uses URIs, which are *universal global unique identifiers*. By using URIs, browsers and other applications can retrieve data of some resource identified by that URI (such a process of retrieving is known as URI dereferencing). Browsers render the retrieved representation so that it can be perceived by a user. Other applications can use the *content negotiation mechanism* to access data in the intended format. URIs of different RDF data can be easily linked. The large-scale, influential initiative known as „Linked Open Data" provides methodology of publishing and interlinking data. Above we mentioned that by linking data we can reuse external resources which we need for our representation (e.g. ontology of ISO 3166 codes for countries). But linked data also allows for gaining new information and formulating interesting queries. For instance, in the Common Data File Format we find references to Countries and Subdivisions. Connecting the countries from the data with e.g. http://www.geonames.org/countries/ automatically gives us information about capital, area or population of a country. So we may form such a request:

„Find a number of legal entities that possess LEI codes for each Country. Order answers by area of a country".

RDF with ontologies also allow for inference. Reasoners for ontologies (e.g. Pellet or HermiT) can check consistency of ontology, i.e. whether the set of ontological restrictions (axioms) has a model. One can also verify whether the facts explicitly expressed in an OWL ontology do not brake the restrictions.

RDF/OWL MODEL OF THE COMMON DATA FILE FORMAT - GENERAL LEGAL ENTITY IDENTIFIER ONTOLOGY (GLEIO)

MakoLab created a RDF/OWL model compliant with the Common Data File Format. The process of development involved identification of:

- classes of objects (e.g. Legal Entity, Business Register);
- individuals (e.g. ISO 3166 codes for countries and subdivisions, particular values of the LEI registration status such as "Annulled" or "Duplicate")
- object properties that relate objects with objects (e.g. hasAssociatedEntity linking concrete legal entities with other legal entities)
- data properties relating objects with literals such as strings, integers or dates (e.g. hasGLEICode linking general LEI registration with its code; it is worth noting that it

was possible to restrict the code of the general LEI to 20 characters)

- constraints that specify the conceptualization of the Common Data File Format standard (e.g. that general LEI registration has at most one LEI code and that each general LEI code identifies exactly one legal entity).

Temporal aspects of LEI data

Temporal aspect plays important role in the Common Data File Format. A general LEI registration status can change in time. The same with data describing legal entities, e.g., the headquarters address of a legal entity can change. GLEIF publishes daily an updated 'concatenated file' which includes all general LEIs issued globally and related LEI reference data published by LEI issuing organizations compliant with the Common Data File Format. In RDF/OWL, we decided to allow for explicit representation of change. Thus we divided all entities into:

- **Variable Entities**, i.e. the entities that have different manifestations at different times.
- **Non-Variable Entities** that do not have different manifestations as different objects at different times. Among non-variable entities are manifestations of variable entities. Each such manifestation has its time stamp (being its last update date).

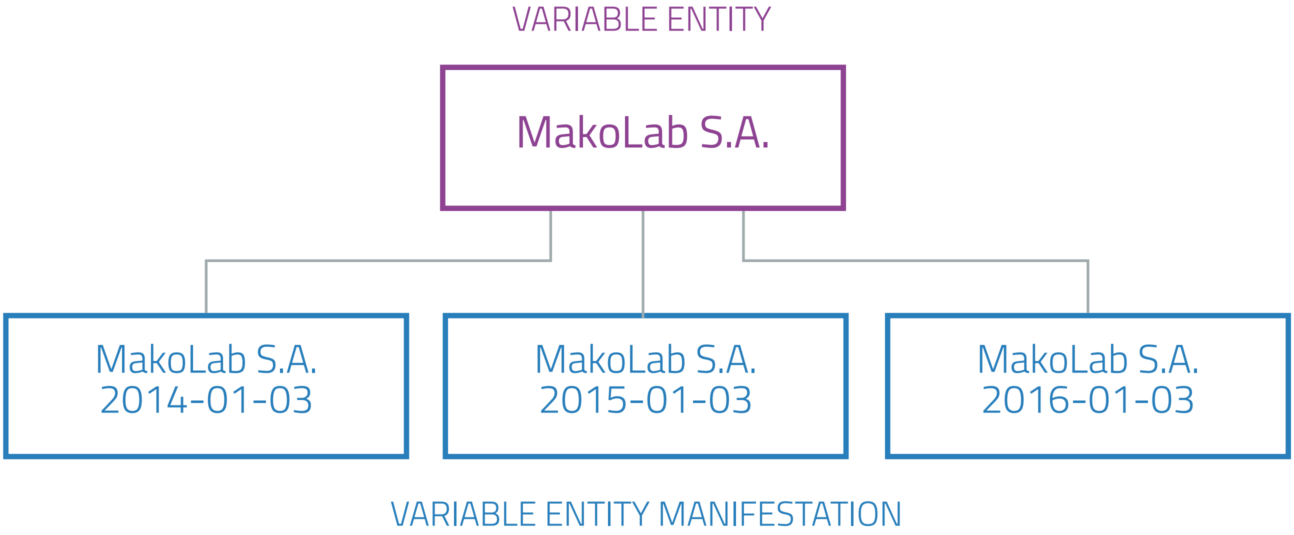


***FIG 2.** Variable entity and its manifestations*

Each Variable Entity has its last (or current if the entity exists) manifestation.
If a variable entity is manifested at time *t* by some entity manifestation *m*, then it exists at *t*, takes a geographical location of *m* and has all qualities possessed by *m* in *t*.

We say that a variable entity possesses a quality *q* (without temporal reference) if all of its manifestations have this quality *q*.

This mechanism/representation allows for change tracking across time stamps.
Every day our LEI application (described in section 3.3 below) reads a complete concatenated XML file and creates new manifestation of a legal entity or global LEI, if a change was found (e.g., LEI status was changed or the headquarters address of the legal entity changed). Manifestations are linked by temporal precedence relation „hasPredecessorEntityManifestation" (see figure 3).

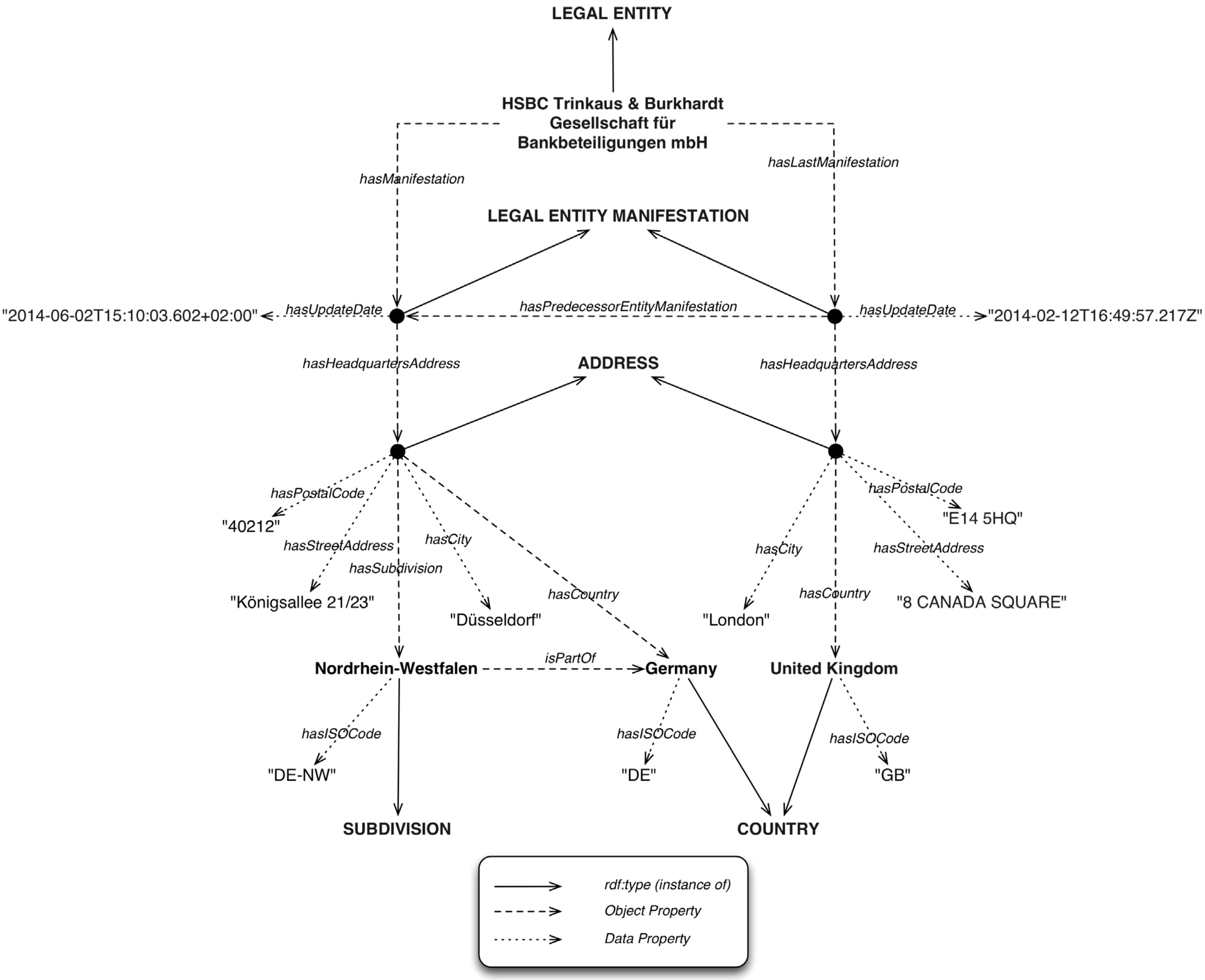


FIG 3. *Legal Entity and its two manifestations. We can observe that the headquarters address of legal entity has changed. Bold upper case names represent classes of object. Black dots and bold lower case names represent individuals. Names in quotes are literals.*

Bold upper case names represent classes of object. Black dots and bold lower case names represent individuals. Names in quotes are literals.

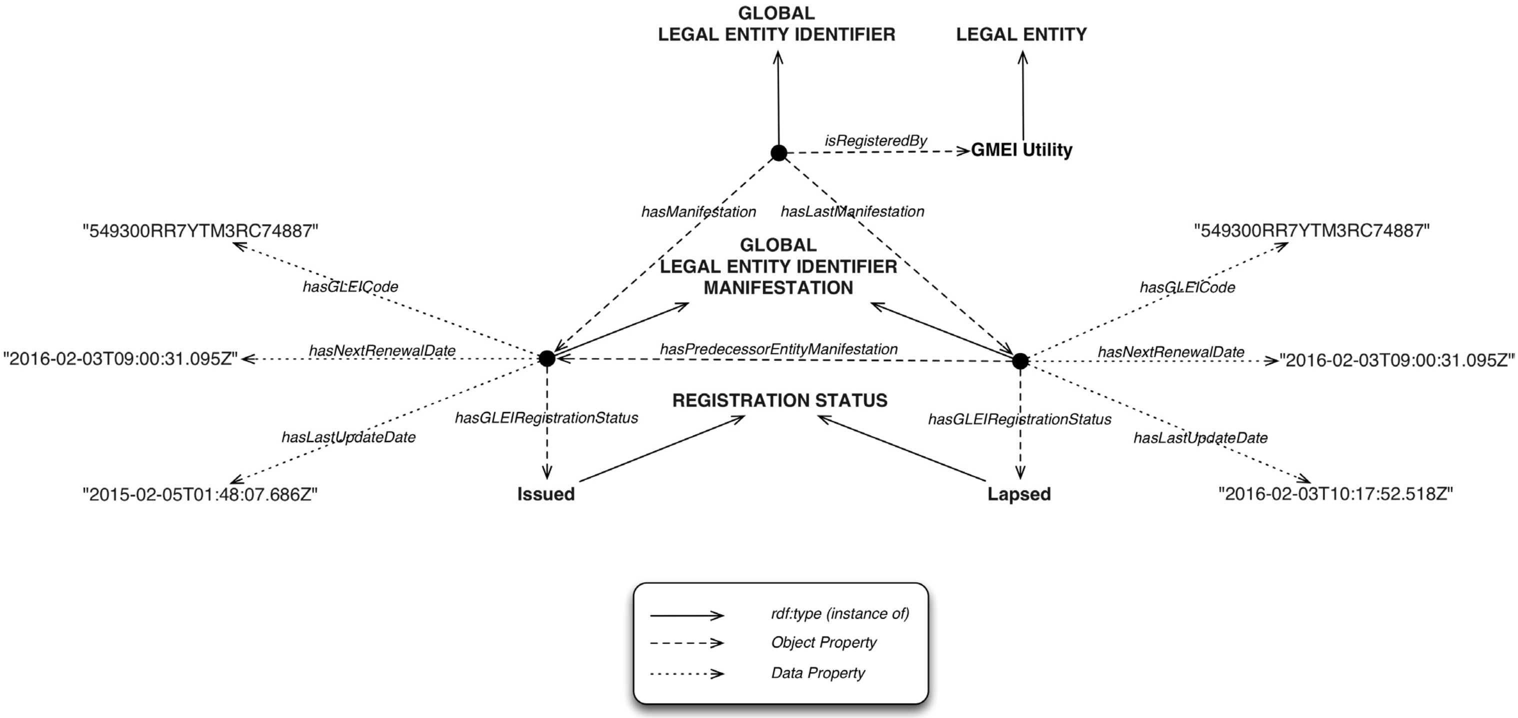


FIG 4. *Global LEI and its two manifestations. We can observe that the global LEI status has changed.*

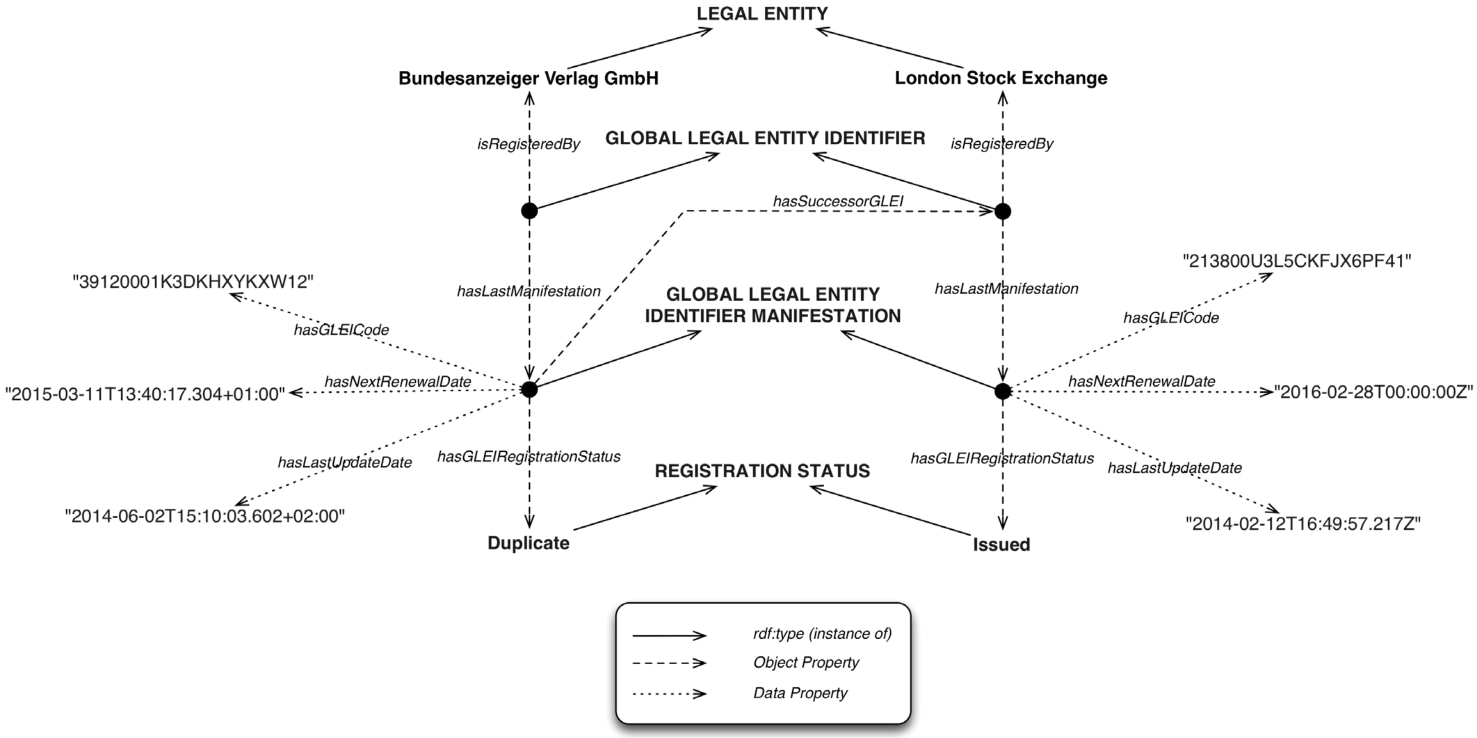


FIG 5. *A global LEI with status „Duplicate" has its successor.*

In figures 4 and 5 above we can see how the global LEI (registry) is represented in GLEIO. For instance, the status of global LEI registration can change from „Issued" to „Lapsed”. But still we have to do with the same global LEI. In that case, we create a new manifestation of the global LEI (see figure 4). In some other cases, for instance when the global LEI status is „Duplicate”, it will have its successor sooner or later. In some sense it ceases to exist and stops being updated. Its successor becomes an issued global LEI (see figure 5).

In figure 6 below we can see how legal entity data is represented in GLEIO. This figure does not take into account a complete description of the global LEI of this legal entity (it may look like above).

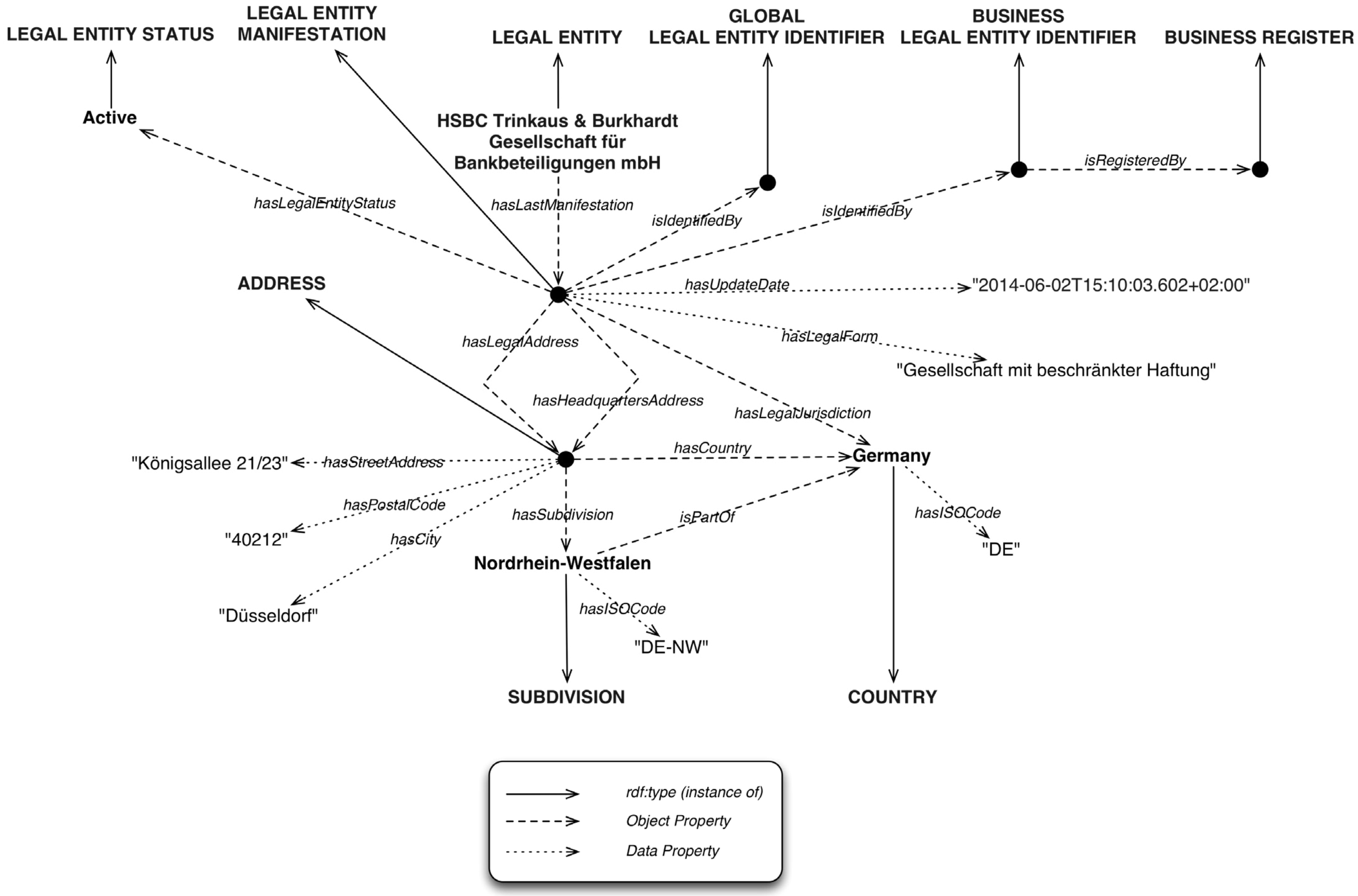


FIG 6. *An example of complete description of legal entity*

WEB PORTAL ALLOWING FOR STORING AND DISPLAYING INFORMATION ABOUT LEI IN LOD STANDARD (LEI.INFO)

GLEIO and Linked Data principles

GLEIO has been used to create a portal allowing for storing and displaying information about LEI. It is complaint with Linked Data principles, i.e.:

1. All entities in GLEIO are uniquely identified by persistent URIs.
2. By using entities' URIs, browsers and other applications can retrieve information about these entities.
3. GLEIO entities are described by natural language annotations describing the meaning as well as by formal axioms constraining the meaning and guaranteeing data coherence and inconsistency.
4. Content negotiation allows to retrieve data adequate to the query. Human agents receive a website, artificial agents get RDF/OWL response.
5. GLEIO can be easily linked with other resources. At that moment, some of the classes and properties are linked with FIBO.
6. SPARQL endpoint allows for querying the global LEI data in a desired way. External application may take advantage of the endpoint and create their own applications reusing data stored by us. Therefore, in order query LEI data or to track changes, one does not have to go through many heavy XML files and compare "strings". It is enough to know the address of the emptying (http://lei.info/sparql) and formulate a proper query.

Components of the web portal

In figure 7 below we present the components of the web portal.

- External XML data - **GLEIF's XML concatenated files** - about entities and identifiers. New data files are published by GLEIF every day.
- **Data transformation component**
  - **Updater** - A console application runs on our server once a day. It downloads a fresh xml data file and checks for newly updated records. For each of them, the updater parses the data, transforms it to an RDF named graph and saves it in a **Triplestore**. The updater is also responsible for tracking changes.
  - **Global Legal Entity Identifier Ontology** provides a schema for the RDF graphs.

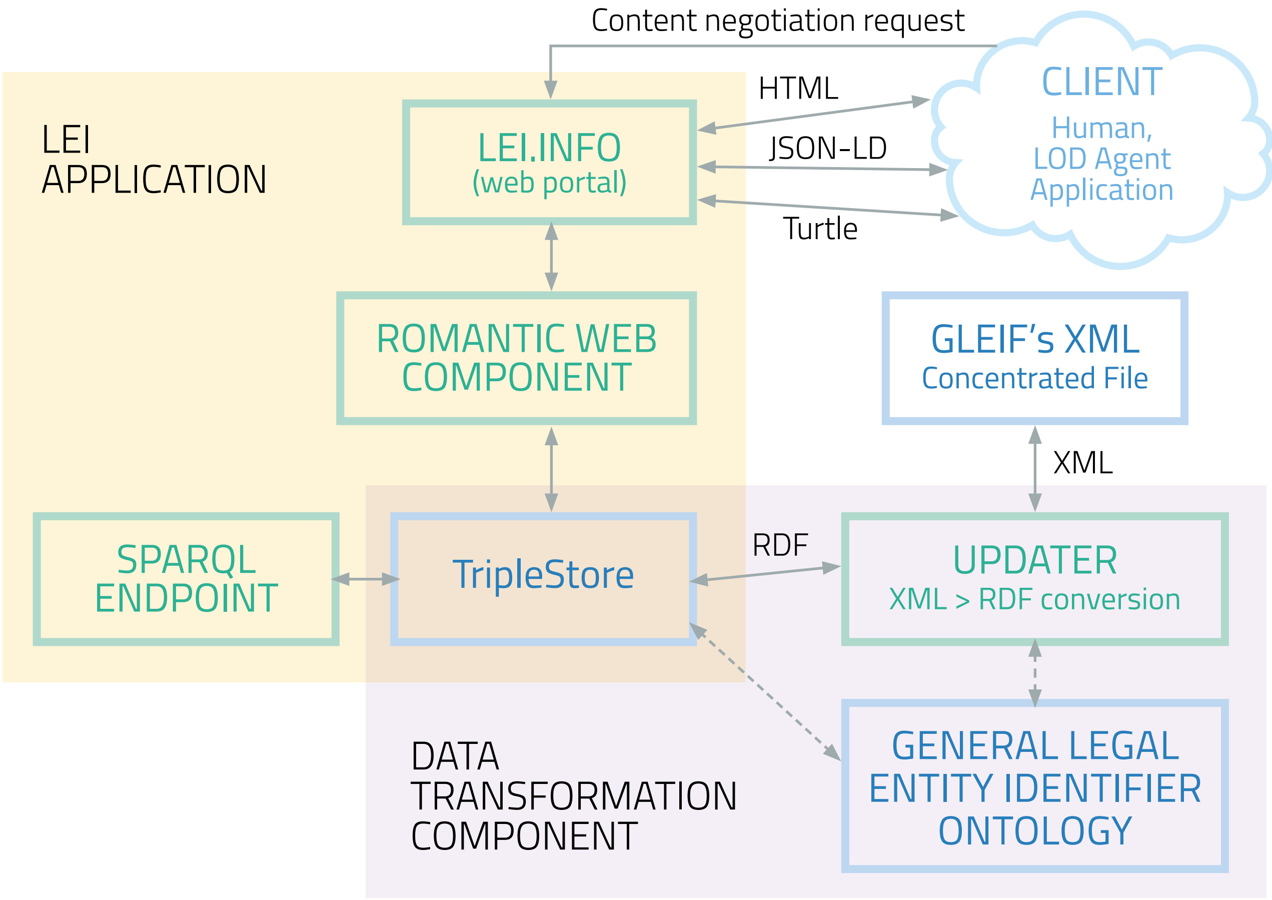

Content negotiation request
LEI APPLICATION
LEI.INFO (web portal)
HTML
JSON-LD
Turtle
CLIENT
Human, LOD Agent Application
ROMANTIC WEB COMPONENT
GLEIF's XML Concentrated File
XML
SPARQL ENDPOINT
TripleStore
RDF
UPDATER
XML > RDF conversion
GENERAL LEGAL ENTITY IDENTIFIER ONTOLOGY
DATA TRANSFORMATION COMPONENT

FIG 7. *LEI application (http://lei.info) and Data transformation component*

- **LEI Application**
  - Global LEI data stored in **Triplestore** can be queried by **SPARQL endpoint**. The endpoint can be used by external applications to access knowledge about LEI data.
  - **Romantic Web** – MakoLab's open source library for .NET used as Triplestore to Object Mapping tool. (http://romanticweb.net/)
  - **lei.info** - Currently it is a website that allows searching legal entities only by LEI or by name and serves entity and LEI info in multiple formats (HTML or many RDF serializations) by content negotiation and RDF embedded in HTML. **In the future it will allow for tracking of changes**.

CONCLUSIONS

We have presented the limitations and problems of the representation of the Common Data File Format in XML and XML scheme. As a solution, we propose a more semantically expressive RDF/OWL model. We could observe the advantages of the semantic representation and its publishing in the Linked Data paradigm.