Financial Regulation Ontology

Tutorial chapter two – loading the law

Importing the text of US laws and regulations into the Financial Regulation Ontology

Loading the law – approach and perspectives

The chapter introduces legislative Use Cases and our approach Semantic ETL. Then we take the less complex source, Code of Federal Regulations (CFR) end-to-end from extract, transformation and load into FRO ontology. Finally we load the United State Code (USC).

Static Reference Data
The Federal Reserve Bank (FED) is the primary regulator for US Banks. The Securities & Exchange Commission (SEC) regulates Investment Funds. US Congress makes the law. We show how LKIF and FRO model these agents, their relationships and actions.

Semantic Integration
Semantic Data Integration is still Extract, Transform, and Load. The tutorial shows conceptual, logical and physical Data Integration model in a semantic environment. We use the TopBraid ontology toolset, but other ontology platforms and RDF data stores can substitute.

XML Source

Ontology Model
This chapter dives deeper into Estrella LKIF modules Action and Medium. We explain where reference and XML source data instantiate classes and properties. FRO extensions to the reference ontology implement a Legal Document structure of defined classes and primitives for CFR and USC.

* See Nuria Casellas for an in-depth introduction and comparison of Legal Ontologies
FRO ties Legal Rules to the text of the law

The **Ontology Knowledge layer** makes the coding logic transparent to provide **Proof** and **Trust**. We want the law and regulations within the ontology and directly link them to their implementation. The diagram shows a Legal Rule, Investor Adviser Act Section §80b-2a 11, the SEC definition of Investment Advisers.

The class **Ex US IAA Section 202_1_11** has the axioms for Investment Advisors to be included under the SEC Rule:

The **owl:equivalentClass** defines then as a UNION of:
- `fo-fr-iaa:Defined_Ex_US_IAA_Section_202_1_11_business`
- `fo-fr-iaa:Defined_Ex_US_IAA_Section_202_1_11_service`
- `fo-fr-iaa:Defined_Ex_US_IAA_Section_202_1_11_compensation`

These are subclasses that in turn define the axioms for the SEC criteria. Likewise the exclusions are encoded as Defined Classes. The Reasoner will evaluate the axiom and place matching Funds as individuals of the defined class. Chapter III – legal reasoning of the tutorial will explain in detail.

The property **lkif-expr:medium** ties the Legal Rule to the USC _Paragraph_ of the Legal Document.

The Paragraph states: “Investment adviser means any person who, for **compensation**, engages in the **business** of advising others...”

Compliance officers and ontology modeler work together translating the legal requirement into a well-defined hierarchy of class definitions. Business Requirements, mapping, business rule logic – **everything is a triple**.
The use case diagram depicts the legislative processes and government actors that FRO is interested in.

Codification and Publication produce the official version of laws and regulation.

- **US Congress** enacts the Law. The original bills passed are input for Rulemaking and Codification. FRO doesn’t hold the bills, but rather revised and codified positive version.
- **Office of the Law Revision Council** codifies the law as an XML available for download.
- The financial regulators, SEC, FED and others are authors of Code of Federal Regulations.
- **Government Publishing Office** make the Code of Federal Regulations available in XML.

The next pages show how FRO captures government actors and processes as static reference data.
US Congress and Lawmaking in the ontology

LKIF and FRO ontology classes hold the Lawmaking use case. The diagram shows instance at the left and their classes, (rdf:type property) to the right. The Supervisory mandate connects to the SEC.

The United States Congress is a LKIF Legislative Body.

The LKIF Act of Law is the passing of a bill. That is when Dodd Frank and the Investment Adviser Act became Law.

The Act of Law authorizes the Securities and Exchange Commission (SEC) to supervise Advisers (the Investment Funds). The FRO Supervisory Mandate is a subclass of LKIF Mandate, with a Regulatory Authority (the SEC) as an actor.
Regulators and Rulemaking in the ontology

LKIF and FRO ontology classes hold the details of the Rulemaking use case. The diagram shows instance at the left and their classes, (rdf:type property) to the right. CFR Title 17, Rule and Regulations, Investment Adviser Act holds the actual text.

The Securities & Exchange Commission is a FRO Regulatory Authority, a subclass of FRO Executive Body, which is a subclass of LKIF Public Body. The SEC is lkif:actor of the Rulemaking.


The Rulemaking creates a LKIF Definitional Expression. LKIF differentiates Expressions from the Process.

The LKIF Definitional Expression bears a Medium. The medium is the actual document. CFR Annual Edition 2016 Title 17 is the FRO Edition. An update of the rules will have a new instance of the edition.
Codification of the Law in the ontology

“Positive law codification by the Office of the Law Revision Counsel is the process of preparing and enacting a codification bill to restate existing law as a positive law title of the United States Code.”

FRO has the Office of the Law Revision Council (OLRC) has a LKIF Legislative Body, because it works for Congress.

“USC_114-219” is the FRO Codification process, a subclass of LKIF Public Act.

The Definitional Expression US Advisor Act is created by the codification. The expression is held by OLRC thus in turn by Congress.

The medium that bears the expression is the Release Point 114-219 of 29 July 2016. FRO Edition is subclass of LKIF Statute, which is subclass LKIF Legal Document “usc-t12-ch17:id81a43...” is the instance of the FRO USC Title, root of over 1,100 components with the text of the law.
The Government Publishing Office (GPO) makes regulations and laws available to the public. The design pattern follows Lawmaking, Rulemaking, and Codification.

The GPO is a FRO Government Office, a subclass of LKIF Executive Body and sibling of FRO Regulatory Authority.


LKIF Definitional Expression was created by the SEC Rulemaking Process we looked at before. It is the resource for the Government Publication. Both are linked to the same version of the text, CFR Annual Edition Title 17.

Note: GPO also publishes PDF and text versions of the United States Code. The Office of the Law Revision Council publishes the XML version in addition. For purposes of Financial Regulation we don’t need to model all intricacies of the Use Cases. Just the context of our XML sources.
The SEC Mandate in context

The diagram shows LKIF instances to define semantics of the IAA legal background. Congress enacted (actor) the 1940 Investor Adviser Act and the Dodd-Frank Act. There are two instances each: The Act and the Statute. The Statute bears the text of the Act of Law passed in congress.

The laws contain provisions that give the Security and Exchange Commission a mandate to supervise Investment Advisers. The SEC enacted a Rulemaking (IAA amended P.L. 112-90). In other words, the SEC announced the final version the regulation. The US Investment Advisers regulation bears the text of the rules (e.g. CFR-2012-title 17-vol3 part 275).
Main FRO/LKIF classes in context

The graph shows some of the LKIF and HFR classes for the instances. Solid yellow dots represent primitive classes.

- Existential ‘some’ restriction
- Universal ‘only’ restriction

Legislative Body is the class for US Congress. The Executive Body holds the SEC and other Regulatory Authorities. An existential restriction ties the Legislative Body to the Act of Law. The US IIA and Dodd-Frank are acts of law. The Act of Law contains a Supervisory Mandate. A class restriction refers to the Executive Body that got the mandate. Corresponding to the Act of Law we have a class for the Rulemaking. The Regulation class anchor for rules.
United States Congress enacted the Investor Adviser Act in 1940 to monitor and regulate the activities of Investment Advisers. The act placed mutual funds, closed-end funds, unit investment trusts, and exchange-traded funds under SEC regulation and supervision. The 2010 Dodd Frank Act (DOF), Title IV required most Hedge and Private Equity Funds to register with the SEC. Full text: https://www.sec.gov/about/laws/iaa40.pdf

The Office of the Law Revision Counsel (OLRC) of the US House of Representatives codifies and publishes the United States Code. As Positive Law the OLRC provides the latest version of the act including all changes and amendments. The OLRC website has the current laws available for download on their website: http://uscode.house.gov/download/download.shtml

FRO uses the OLRC XML Title 12 and 15 as a data source.

The Security and Exchange Commission implements the law. The SEC revises the Code of Federal Regulation with detailed instructions, forms, and procedures. The SEC hands over the new CFR to GPO for publication. The SEC supervises Investment Companies and Advisers. Note: For Banking the Federal Reserve is the main regulator.

The Government Publishing Office is the official source for Federal Government information. The “bulk” data is available in on the Federal Digital System (FDSys).

The XML schema and a user guide are also available. http://www.gpo.gov/fdsys/bulkdata/CFR/resources/CFRMergedXML.xsd

FRO uses GPO XML files as a source for CFR.
CFR and USC in FinRegOnt ontology files

Financial Regulation Ontology instance files contain the text of laws & regulations relevant to Investment and Banking.

<table>
<thead>
<tr>
<th>Title</th>
<th>Chapter</th>
<th>United States Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 Banks and Banking</td>
<td>17</td>
<td>Bank Holding Companies</td>
</tr>
<tr>
<td>12 Banks and Banking</td>
<td>53</td>
<td>Wall Street Reform and Consumer Protection</td>
</tr>
<tr>
<td>15 Commerce and Trade</td>
<td>275</td>
<td>Investment Companies and Advisers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title</th>
<th>Chapter</th>
<th>Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 Banks and Banking</td>
<td>II</td>
<td>Federal Reserve System</td>
</tr>
<tr>
<td>17 Commodity and Securities Exchanges</td>
<td>II</td>
<td>Securities and Exchange Commission</td>
</tr>
<tr>
<td>15 Commerce and Trade</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

http://finregont.com/fro/usc/
FRO_USC_Title_12_Chapter_17.ttl
FRO_USC_Title_12_Chapter_53.ttl
FRO_USC_Title_15_Chapter_2D.ttl

http://finregont.com/fro/cfr/
FRO_CFR_Title_12_Part_217.ttl
FRO_CFR_Title_12_Part_225.ttl
FRO_CFR_Title_12_Part_252.ttl
FRO_CFR_Title_17_Part_275.ttl

This section describes the architecture moving Legal Sources into the Financial Regulation Ontology. The Data Integration process is similar to traditional Warehouses. We adopt the Giordano’s integration modelling approach (Anthony David Giordano Data Integration Blueprint and Modeling, IBM Press 2011).

### High Level Conceptual Semantic Integration Model

The “database” symbols stand for persistent storage in general. The rectangles denote a process.

Data sources can have various formats:
- Ontology files,
- XML,
- Spreadsheets,
- RDF Databases,
- any data source with JDBC connectivity.

We use TopBraid Maestro to import XML, but the Protégé and RDF Database environment also provide imports.

RDF Staging and Target Ontology can be OWL files or graphs in a RDF Database.

The Ontology Extract imports the Data Source and stores it in a “dumb” RDF Staging representation.

The Ontology Transformation operates completely in the Semantic environment. The transformation logic is encoded in SPARQL rules.

We use TopBraid SPIN, but Protégé and RDF Database environments also support SPARQL based rules.

RDF Staging is critical to the architecture and should not be bypassed. We do not want to encode business logic in the Extract process.

We want uniform staging classes and utilize semantic transformation not matter what the data source is.
Seminar integration Logical Model

United States Code

Title 12  
Chapter 17  
Chapter 53  
Title 15  
Chapter 2D

Extract the USC source data. Import into United States Legislative Model (USLM) classes.

Code of Federal Regulations

Title 12  
Part 217  
Part 225  
Part 252  
Title 17  
Part 275

Extract the CFR source data. Import into Federal Digital System (FDSys) classes.

FRO RDF Staging

USC_USLM_Schema
uslm:title
uslm:chapter
uslm:section
uslm:paragraph
uslm:note

CFR_FDSysElement
cfr-fdsys:s:TITLE
cfr-fdsys:s:PART
cfr-fdsys:s:SECTION
cfr-fdsys:s:P (paragraph)

Transform USLM/FDSys staging instances. Load the information into ontology classes.

The inference engine executes rules attached to the source class: Mapping rules perform simple triple movement. SPARQL rules CONSTRUCT complex transformations.

Financial Regulation Ontology

FRO Legal Reference
Document Component
Title
Chapter
Chapter Division
Section
Paragraph
divides
refers to
Heading
Subject
Text
Identifier
While the Logical integration model referred to persistent storage, business concepts and ontology classes, the Physical Load Models names the websites and files.

We download CFR XML files and schema from the GPO website, bulk data directory. The ontology editor imports the XSD and converts XML to a Semantic view. We export the both RDF Graphs to target OWL (turtle) files.

For staging we want a single OWL file with class definitions for the CFR concepts (Section Paragraph Note etc.) All four instance data files import the common class definitions.
Understanding the CFR XML Source

We follow “Private Fund Exception” in section 203 of CFR-2012-title17-vol3-part275 from XML to FRO ontology.

The GPO website provides the XSD-Schema and documentation
- CFRMergedXML.xsd

The XSD Schema is generic for all Federal Regulations. A good example of the Semantic Web Layers discussed previously. We get the machine readable syntax, but no semantics for the particular regulation.

The header contain identifying information about the document

The regulation text is structured in sections. <SECTION>
- Section number with the paragraph reference <SECTNO>
- Subject of the section <SUBJECT>
- And individual paragraphs <P>.

We follow “Private Fund Exception” in section 203 of CFR-2012-title17-vol3-part275 from XML to FRO ontology.

The GPO website provides the XSD-Schema and documentation
- CFRMergedXML.xsd

The XSD Schema is generic for all Federal Regulations. A good example of the Semantic Web Layers discussed previously. We get the machine readable syntax, but no semantics for the particular regulation.

The header contain identifying information about the document

The regulation text is structured in sections. <SECTION>
- Section number with the paragraph reference <SECTNO>
- Subject of the section <SUBJECT>
- And individual paragraphs <P>.

We follow “Private Fund Exception” in section 203 of CFR-2012-title17-vol3-part275 from XML to FRO ontology.

The GPO website provides the XSD-Schema and documentation
- CFRMergedXML.xsd

The XSD Schema is generic for all Federal Regulations. A good example of the Semantic Web Layers discussed previously. We get the machine readable syntax, but no semantics for the particular regulation.

The header contain identifying information about the document

The regulation text is structured in sections. <SECTION>
- Section number with the paragraph reference <SECTNO>
- Subject of the section <SUBJECT>
- And individual paragraphs <P>.
Generating OWL classes from CFR XSD

TopBraid Composer Maestro Edition (TBC) is our main ontology editor.

1. First we import the XSD-Schema. There are various options to customize the import. The tool creates on OWL file CFRMergedXML.ttl with classes for the XSD elements. We open CFR-2012-title17-vol3-part275.xml with TBC’s “Semantic XML”. This show instances of the classes with the actual XML text fragments. Note that this is only a semantic rendition. The underlying file is still XML.

2. Finally we export the graph to create save it as OWL. This is for convenience and performance, so we don’t have to repeat the import steps. We use the same filename, but with extension TTL: CFR-2012-title17-vol3-part275.ttl
The ontology browser shows CFR-2012-title17-vol3-part2752.ttl Classes and Properties.
We create the namespace “cfr-fdsys-s”. (showing left of the colon on the classes/properties).
We use a collection class, CFR_FDSysElement as a superclass for all CFR classes.

The import creates 3 properties (prefix “composite”, preserve the XML structure:

- Object property “composite:child” denotes that a domain instance is underneath the range. For example “Part 275” has composite:child Section “§ 275.204-5”.
- Object property “composite:parent” denotes the inverse.
- Data Property “composite:index” adds a sequence number to child elements. So we can preserve an query the order of paragraph within a Section.
Comparing the XML to OWL instance and class structure

The diagram shows the “Private Fund Exception”, 203 of CFR-2012-title17-vol3-part275 graph.

The XML `<SECTION>` becomes an instance of `cfr-fdsys-s:SECTION` class.

The XML elements within the section are generated as instances connected with the composite:child object property:

Every section has a section number, `cfr-fdsys-s:SECTNO` and subject, `cfr-fdsys-s:SUBJECT`.

Paragraphs become instances of `cfr-fdsys-s:P`

Object property composite:child connects the instances.

```xml
<SECTION>
  <SECTNO>$ 275.203(m)-1</SECTNO>
  <SUBJECT>Private fund adviser exemption.</SUBJECT>
  <P>(1) Acts solely as an investment adviser to one or more qualifying private funds; and</P>
  <P>(2) Manages private fund assets of less than $150 million.</P>
</SECTION>
```
We can introspect all details of “Private Fund Exception” in the Form tab.

The top of the window shows path and name of the open OWL file.

FinRegOnt sets the prefix as cfr-17-275, title and part of the code of federal regulations. The import generates a URI for the instance: cfr-17-275:r-0-9. “r” denotes resource. The numbers are generated based on the hierarchy level. We use an import generated URI, because the CFR XML doesn’t come with an ID element. For United States Code we use the element Identifier in the XML.

The composite index means that section “§ 275.203(m)-1” is the 17th child element under the parent (Part 275).

Under composite:child we find section number, subject and paragraphs. We expand paragraph (1) – “Acts solely as an investment adviser to one of more private funds ....”

Again, the child elements have a composite index. We use the index number to query the paragraphs in the right order.

All text fragments are stored as sxml:TextNode, a generic Semantic-XML class with text and composite index. We use will the index number to concatenate the text fragments into the full text of a paragraph.

To use the data warehousing ETL analogy, this is just an Extract of into simple Staging classes. rather than ontology design. There are no semantics and it is hard and error prone to SPARQL query the data.

We will transform the staging structure and load into LKIF classes. First we revisit the reference ontology and extend the design. Then we use SPIN rules to move the instances.
CFR to FRO/LKIF Physical Transform Model

The Physical Load model shows the source files and target files in the Financial Regulation Ontology.

Prerequisite of the Semantic ETL is a mapping from CFR_FDSys_Schema.ttl to the target classes in Code_Federal_Regulations.ttl. The target ontology imports US_Legal_Reference.ttl, an ontology common to CFR and USC and Legal_Reference.ttl, common to international regulations. Legal_Reference.ttl imports LKIF.

We create a target instance ontology file for very staging file. E.g. FRO_CFR_Title12_part_275.ttl for CFR-2012-title17-vol3-part275.ttl

The target files only contain class and property instances. They all import the common schema, Code_Federal_Regulations.ttl
Identifying and extending the LKIF target for CFR

We extend the model of the US Legal Context, described in previous slides to accommodate the CFR elements.

Slide SEC Mandate in context defined the SEC as a LKIF Executive Body and Financial Regulation as an outcome of its Rulemaking. fro-cfr:CodeFederalRegulations is a subclass of Financial Regulation. CFR_Component is a collection class for all CFR element classes.

The top class, CFR_Title is also member of the Document Edition, described in Codification of the Law in the ontology.

The object property fro-leg-ref:divides creates the hierarchy of CFR_Components. Paragraphs divide the Sections. Section divides the Part. Part divides the Chapter and finally Chapter divides the Title. We create a class restriction: CFR_Paragraph divides some CFR_Section.

Next pages show, how to transform the staging data and load into LKIF classes. First we look at ETL in the Semantic world. How to use SPARQL and TopBraid SPIN rules to move the instances.
The CONTRUCT statement is the equivalent of the INSERT in a relational database. We use SPARQL CONTRUCT statements to load the data into our target classes.

“The CONTRUCT query form returns an RDF graph. The graph is built based on a template which is used to generate RDF triples based on the results of matching the graph pattern of the query.” (W3C SPARQL Query Language for RDF)

The example creates triples (an RDF graph) for the hasParagraphText property. We have two variables:

• ?targetInstance of our destination fro-cfr:CFR_Paragraph
• ?text The actual text that we and to construct for the CFR_Paragraph instance.

CONSTRUCT {
  ?targetInstance fro-cfr:hasParagraphText ?text .
}
WHERE {
  ?this composite:child ?text_node .
  ?text_node a sxml:TextNode .
  ?text_node sxml:text ?text .
}

The WHERE clause specifies the result set from our Source ontology classes. This is similar to SQL “INSERT ... AS SELECT”. ?this is a special variable that refers to the current instance of this class that is being evaluated by the inference rule. For every instance of our source “cfr-fdsys-s:P” we navigate to “sxml:TextNote” and assign the “sxml:text” property to our ?text variable. The BIND keyword assigns the value of an expression to a variable. In this case, we call a function to convert the URI of ?this to the target URI.
SPIN – SPARQL Inferencing Notation

With TopBraid Composer as ontology editor, we use SPARQL Inference Notation, SPIN to define mapping rules.

“SPIN is a W3C Member Submission that has become the de-facto industry standard to represent SPARQL rules and constraints on Semantic Web models. SPIN also provides meta-modeling capabilities that allow users to define their own SPARQL functions and query templates. Finally, SPIN includes a ready to use library of common functions.” (http://spinrdf.org/)

“SPINMap is a SPARQL-based language to represent mappings between RDF/OWL ontologies. These mappings can be used to transform instances of source classes into instances of target classes.”

There are no mapping spreadsheets or proprietary ETL files.

The mapping file is in Ontology Web Language. That means, we can query target schema and data joined with the mapping to their source.

We create a new RDF/SPIN mapping file, CFR17_275spinFRO.ttl in TopBraid Composer. The mapping file imports the required SPIN MAP elements, to support mapping rules and editor.

Next we import the source ontology, CFR-2012-title17-vol-3-part275.ttl and target, FRO_CFR_Title_17_Part_275.ttl

Now we have visibility of source and target classes in our mapping ontology. We invoke the mapping editor.
SPIN – SPARQL Inferencing Notation

With TopBraid Composer as ontology editor, we use SPARQL Inference Notation, SPIN to define mapping rules.

The diagram shows the CFR Paragraph mapping from staging, cfr-fdsys-s:P to target fro-cfr:CFR_Paragraph:

We connect the two classes with a “change namespace” mapping rule. For every source instance, this will create a target instance with the URI namespace http://finregont.com/fro/cfr/FRO_CFR_Title_17_Part_275.ttl

The second rule copies data property composite:index to the target fro-leg-ref:hasSequenceNumber.

The mapping contexts for CFR Paragraph, Section and Note are stored as an instance of spinmap:Context.
Connecting a source to a target class invokes the mapping rule dialog.

The Target Function specifies how to create the target URI.

For CFR Paragraph we simply change the namespace.

The Preview window shows how source instances are converted to target URI. The Result column shows the prefix frcfr-t17-p275 and instances r-33-20-2, r-1-25-57.

The SPARQL Expression is the WHERE clause for the CONSTRUCT statement.
- The first BIND extracts the local name (right of the colon) of the source URI.
- The second BIND concatenates the local Name to the target namespace.
SPIN rules populating properties (1)

Once defined, the mapping context will be used to populate data and object properties.

The spinmap-rules can be examined and customized at the source class’ Form tab. The inference engine (reasoner) will trigger the rules for every instance of the class. The rules “Map into” the mapping context with a “derive” rule: Derive the target fro-leg-ref:hasSequenceNumber from source composite:index, cast the value to xsd:integer.

Likewise spin-rules show at the source class’ Form tab. Spin-rules are free form SPARQL that we the inference engine to execute. In this case we want to CONSTRUCT the fro-cfr:hasParagraphText data property. The WHERE clause navigates from the ?this instance to the sxml:TextNode. The BIND statement assigns the target instance using the Mapping Context.
Templates facilitate reuse of common SPARQL rule statements.

All FinRegOnt instances have an object property to point to their source instance:

```
fro-leg-ref:hasSourceInstance.
```

The **CONSTRUCT** sets the domain target instance and ?this variable range.

The **WHERE** clause assigns the ?targetInstance variable.

- The first **BIND** extracts the local name (right of the colon) of the source URI.
- The second **BIND** concatenates the local Name to the target namespace.
Running the Inference Engine

The inference engine executes SPIN-rule alongside standard reasoning.

An inference engine is a software able to infer new facts from a set of asserted facts and rules. Non-ontology Inference engines are also to derive decision based on business rules (rules engine). The reasoner generalizes the concept with a richer, ontology based semantic.

• Input Asserted Triples are facts in the included staging file CFR-2012-title17-vol-3-part275.ttl.
• The TopSPIN (SPARQL Rules) are in the mapping file: CFR17_275spinFRO.ttl

We run the engine from the TopBraid composer Menu or Button. The engine iterates through standard reasoning for class subsumption. That is to infer that an instance must be type of a class based on its asserted properties. Chapter I of the tutorial touched how the defined class drives the reasoner. Chapter III will explain the central role of reasoning for financial compliance.

The TopSPIN engine will also execute the SPIN rules. The status window shows rule “STEP001: set Paragraph Text” on cfr-fdsys-s:P
For all instances of cfr-fdsys:s:P the engine will execute the SPARQL and CONSTRUCT output triples. The iteration (7) means that this is the seventh pass of the engine. As configured the reasoner will iterate until there are no more new triple inferred.
Inferencing Output Triples

The Inferences tab shows the output triples in three columns, subject, predicate, and object.

We scroll down the Subject column fro-cfr-t17-p275:r-1-17-3, the Private Fund Adviser paragraph and Look at Predicate column for object and data properties of the paragraph:

- Object Property hasSourceInstance, links to the CFR FDSys original (Object column).
- Paragraph text is sub property of Component Text. The engine infers a triple for the component text.
- It has a sequence number: 3
- It has a text: “Acts solely ....”
- The paragraph is a fro-cfr:CFR_Paragraph

30519 triples added
Validating target instance – class browser

Once the inferencing is complete, the class browser will indicate the number of triples next to the Code of Federal Regulations Classes.

This is the first consistency check. The number of target instances should match the number of elements in the CFR XML. Title 17 Part 275 has 2 Chapters, 44 Sections, 757 paragraphs and 16 Notes. We select CFR_Section and the instances tab shows the individual sections.

Scope:
FinRegOnt does not import everything from the CFR XML source. For our purpose of Legal Reasoning, we are interested in the text and structure of the regulation. The lowest level is the paragraph or note that we want to link via object properties to a Legal Expression. We do not need the XML table of contents and formatting. However, the hasSourceInstance property links to the source instance, where all details remain available for querying.
Validating target instances – Resource Form

The Resource Form displays all details of a class instance. This is the next step in checking consistency of the rule output. We double click on Section 17, § 275.203(m)-1 in resource list to launch the resource from.

The class instance has a citation, [76 FR 39703, July6, 2011]. Note that SECTION has a SPIN-rule to populate the citation from its composite:child cfr-fdsys-s:CITA.

The section number § 275.203(m)-1, populated from cfr-fdsys-s:SECTNO

The section subject “Private fund adviser exemption, populated from cfr-fdsys-s:SUBJECT

Section 17 is divided by many paragraphs (only 3 are displayed here). A rule on SECTION constructs from SECTION composite:child with type of cfr-fdsys-s:P

The section divides Part 275 of the Code of Federal Regulations. The construct is static, because there is only one CFR Part in our source XML.

The component name is parent property of section subject. Siblings are Part text, Volume text, Chapter text and Title text. This facilitates queries across CFR Components. The value is inferred automatically. We do not need a rule.

A rule sets the source instance to the original cfr-fdsys-s:SECTION resource.

Section 17 has a note. The object property fro-leg-ref:refers_toNote links to the fro-cfr:CFR_Note instance.

The FinRegOnt documentation has definitions for all classes and properties on the website: http://finregont.com/ontology-documentation/
Validating target instances – the Graph

The Semantic Data Management provides complete traceability and linage. Everything is a triple. We can navigate and query data instance, Section § 275.203(m)-1 to Source, Mapping and Reference Data.

We start with the Section fro-cfr-t17-p275:r-1-17, “Private fund adviser exemption”. The object property fro-leg-ref:divides navigates to the higher CFR components, Part, Chapter and Title.

Regulation instance fro-cfr:CFR_Title-17 has object properties to the Reference Data. Title 17
• is a lkif-mero:member_of the CFR edition.
• lkif-mero:bears the Securities & Exchange Commission expression of the regulation.

The two anchor points let us query all information described in the Legislative Contact.

The section instance has the object property fro-leg-ref:has SourceInstance pointing to its original Section. With it we can join to all imported CFR FDSys classes and properties.

The rdf:type of the section instance is spinmap:targetClass for the mapping. From here we navigate to the spinmap:sourceClass.
Querying the meta data

We can query the graph joining our section with reference-, source, and mapping data.

SPARQL query
The query traverses the complete meta-data graph, starting with the section. Variables “?” and object properties perform the joins.

```sparql
SELECT *
WHERE {
  BIND (fro-cfr-t17-p275:r-1-17 AS ?froSection) .
  #reference data
  ?froSection fro-leg-ref:divides ?froPart .
  ?froChapter fro-leg-ref:divides ?froTitle .
  # source
  ?froSection fro-leg-ref:hasSourceInstance ?source .
  ?cfrNumber a cfr-fdsys-s:SECTNO .
  # mapping
  ?froSection a ?froTargetClass .
  ?spinContext spinmap:targetClass ?froTargetClass .
  ?spinContext spinmap:sourceClass ?cfrSourceClass
}
```

Query Result Set
The query is a star “*”, so all query variables show in the result set. The query can be customized to include more or even all sections. (modify or omit the `BIND` statement).

Because everything is a triple within the ontology, we have a whole meta-data repository at hand.
Asserting the mapped triples

The populated FRO instances are still inferences. To make them permanent, we assert them to the target file.

Dynamic/volatile Data should be inferred at runtime
An ontology that needs the populated sections and paragraphs can simply import the mapping file. That means the mapping rules will be executed every time we invoke the reasoner. This is desired for volatile information. The triples are temporary and ‘lost’ when we close the file. Dynamic strategies keep the target ontology in sync with the source. The cost is computing time of the inference engine.

Static and slowly changing data should be asserted
For data that does not change often, we don’t want to spend reasoner time. This pertains to static reference data, transactions, and history. The reasoner does not have to re-compute the transformation rules. The triples are permanent. This means that changes to the source are not reflected in the target ontology automatically.

For CFR we assert the triples to a new file: FRO_CFR_Title_12_Part_275.tll

All ontology environments have routines to export inferences, results to a new graph. A challenge is to separate wanted from unwanted inferences. See our earlier example, fro-cfr-t17-p275:r-1-17-3, the Private Fund Adviser paragraph:

- We only want the rules output, new FRO class instances and their properties.
- We do not want derived inferences, like subsumption to parent classes and properties. They should be inferred in the target ontology, because the target schema may change.

For the Financial Regulation Ontology, we explicitly move triples using SPARQL Motion.
SPARQL Motion – scripting language

“SPARQL Motion is an RDF-based scripting language with a graphical notation to describe data processing pipelines.”

“The basic idea of SPARQL Motion is that individual processing steps can be connected, so that the output of one step is used as input to the next. RDF graphs are the basic data structure that is passed between the steps, but named variables pointing to RDF nodes and XML documents can also be passed between steps. The behavior of each module is typically driven by SPARQL queries, for example to iterate through result sets, to construct new RDF triples and to perform updates to RDF data sources.” [http://sparqlmotion.org/](http://sparqlmotion.org/)

SPARQL Motion is quite powerful and flexible – similar to ETL environments. The diagram shows a subset of the script to load the CFR rule inferences.

Input for the script is an import of the CFR mapping file with inferences.

RDF Processing elements are SPARQL CONSTRUCT statements that operate on the input triples. For example:

CONSTRUCT {
  ?section a fro-cfr:CFR_Section .
}
WHERE {
  ?section a fro-cfr:CFR_Section .
}

The constructed triples become input for the next step.

The final Export step specifies to write the output to [FRO CFR Title 17 Part275.ttl](http://finregont.com)
Querying the Code of Federal Regulations

The CFR “everything query” contains the main Code of Federal Regulations classes and data properties. We use the query to validate the data import for FRO resource files.

The select joins the section with
- Reference data
- Section sequence number, source instance and subject.
- Section Notes and Citation
- Paragraph sequence number and text and enumeration text.

We sort by section sequence than paragraph sequence.

The query selects (almost) everything in FRO related to the Code of Federal Regulations. Besides validation, we run the query to export data into csv/MS-Excel format. Query file and Excel are in the website directory: http://finregont.com/fro/query/
United States investment advisers.

(a) For purposes of section 203(m) of the Act (15 U.S.C. 80b-3(m)), an investment adviser with its principal office and place of business in the United States is exempt from the requirement to register under section 203 of the Act if the investment adviser:

(1) Acts solely as an investment adviser to one or more qualifying private funds; and

(2) Manages private fund assets of less than $150 million.

Non-United States investment advisers.

(b) For purposes of section 203(m) of the Act (15 U.S.C. 80b-3(m)), an investment adviser with its principal office and place of business outside of the United States is exempt from the requirement to register under section 203 of the Act if:

(1) The investment adviser has no client that is a United States person except for one or more qualifying private funds; and

(2) All assets managed by the investment adviser at a place of business in the United States are solely attributable to private fund assets, the total value of which is less than $150 million.

Frequency of Calculations.

(c) For purposes of this section, calculate private fund assets annually, in accordance with General Instruction 15 to Form ADV (§ 279.1 of this chapter).

Definitions.

(d) For purposes of this section:

Assets under management means the regulatory assets under management as determined under Item 5.F of Form ADV (§ 279.1 of this chapter).

Place of business has the same meaning as in § 275.222-1(a).

Principal office and place of business of an investment adviser means the executive office of the investment adviser from which the officers, partners, or managers of the investment adviser direct, control, and coordinate the activities of the investment adviser.

Private fund assets means the investment adviser’s assets under management attributable to a qualifying private fund.

Qualifying private fund means any private fund that is not registered under section 8 of the Investment Company Act of 1940 (15 U.S.C. 80a-8) and has not elected to be treated as a business development company pursuant to section 54 of that Act (15 U.S.C. 80a-53). For purposes of this section, an investment adviser may treat as a private fund an issuer that qualifies for an exclusion from the definition of an “investment company,” as defined in section 3 of the Investment Company Act of 1940 (15 U.S.C. 80a-3), in addition to those provided by section 3(c)(11) or 3(c)(7) of that Act (15 U.S.C. 80a-3(c)(11) or 15 U.S.C. 80a-3(c)(7)), provided that the investment adviser treats the issuer as a private fund under the Act (15 U.S.C. 80b) and the rules thereunder for all purposes.

Related person means any person that is a U.S. person as defined in § 230.902(e) of this chapter, except that any discretionary account or similar account that is held for the benefit of a United States person by a dealer or other professional fiduciary is a United States person if the dealer or professional fiduciary is a related person of the investment adviser relying on this section and is not organized, incorporated, or (if an individual) resident in the United States.
United States Code - Physical Load Model

The USC Lead Model mirrors the CFR steps, however the structure of the law is more complex than regulations. There are many more components and variations in the hierarchy.

The Office of the Law Revision Council provides one XML file per USC Title (usc12.xml, usc15.xml). FRO only needs individual chapters - not the whole title. We used XMLSpy to spit the XML into chapter files.

The XSD is called United States Legislative Model (USLM). We adopt the abbreviation for FinRegOnt staging schema.
Understanding the USC XML header

The USC header contains version, the schema include, and metadata about the document.
The import into Financial Regulation Ontology Staging retains this information, but we only need selected elements for LKIF/FRO. The only XMLSpy edit to the was to remove chapters that we don’t need.

The <meta> section has the document title, number.
Publisher is the Office of the Law Revision Council (OLRC).
The Publication name refers to the OLRC release point 114-153.
The files is as of 2016-04-27.

The document body starts with the <main> tag. Just like in CFR, top-level component is the Title. The id field (see title and note) provides a unique identifier for all components, that we will use to construct the URIs in FinRegOnt.
Understanding the USC XML main structure

The USC main structure has more levels and content elements than CFR. Our Private Fund Manager Exception is contained in a Subsection of 15 U.S. Code § 80b–3.

The XML has a rich set of formatting styles and a table of contents. Both is not needed in FRO for our purpose of Legal Reasoning and we don’t import them. They remain available in the staging RDF. We do import:

- **id**
  - The unique element identifier
- **identifier**
  - The human readable index of the element
- **heading**
  - The name/title of the element.

Section § 80b–3, breaks down into 14 subsections. Subsection b/1 defines exception to the registration requirement. The chapeau is an introductory text for the following lower levels. In this case a heading for the 7 paragraphs. Finally the content element contains the text.

```xml
<section style="uslm:lc:I80" id="idd04f5b5e-0c74-11e6-aa53-e455a13f2ad9"
  identifier="/us/usc/t15/s80b-3">
  <num value="80b-3">§ 80b-3.</num>
  <heading>Registration of investment advisers</heading>
  <subsection style="uslm:lc:I19" id="idd04f5b5f-0c74-11e6-aa53-e455a13f2ad9"
    identifier="/us/usc/t15/s80b-3/a">
    <num value="b" class="bold">(b)</num>
    <heading class="bold">Investment advisers who need not be registered</heading>
    <chapeau>The provisions of subsection (a) shall not apply to—</chapeau>
    <paragraph style="uslm:lc:I12" id="idd04f5b60-0c74-11e6-aa53-e455a13f2ad9"
      identifier="/us/usc/t15/s80b-3/b/1">
      <num value="1">(1)</num>
      <content>any investment adviser, other than an investment adviser who acts as an investment adviser to any private fund, all of whose clients are residents of the State within which such investment adviser maintains his or its principal office and place of business, and who does not furnish advice or issue analyses or reports with respect to securities listed or admitted to unlisted trading privileges on any national securities exchange;</content>
    </paragraph>
  </subsection>
</section>
```
Imported USC USLM classes & properties


The XML header elements, creator, publisher, title, and type have been imported as classes with the “dc:” prefix. The namespace refers to http://purl.org/dc/elements/1.1/ the Dublin Core resource metadata standard.

We create the namespace “uslm” (=United States Legislative Model, showing left of the colon on the classes/properties).
We use a collection class, USC_USLM_Schema as a superclass for all USLM classes. More than 10,000 instances have been created.

The properties list starts with object property composite:child/parent and the index - just like the CFR import.
Comparing the USC USLM § 80b–3 graph

The diagram shows the instance graph for the Private Fund Exemption XML.
The instance box text starts with the class name, for example “<uslm:section “. The connecting arrows, object property instances of composite:child show the hierarchy from Section via Subsection to Paragraph.

**Section** and **Subsection** have a heading, “Registration of Investment Advisers”. The “num” instance, child of section has numbering text § 80b–3. Section also shows the **Source Credit** and a **Note**. The Subsection also has a Num and Heading “Investment Advisers who need not to be registered”. The **Chapeau** “Provisions of subsection (a) shall not apply to …” precedes the list of paragraphs. **Paragraphs** have a Num value and content. The Num value helps to list the exceptions in the right order. **Content** can be a simple text or more complex structure as we shall see on transformation and query slides.
A double click on the section instance invokes the details dialog.

The instance is a uslm:section. The uslm:id-section is a unique identifier coming with the OLRC XML. We will use this ID to generate the FinRegOnt URL. The uslm:identifier-section is the human readable ID. It is structured as an index into the United States Code.

We ignore the uslm:style-section. The formatting can be retrieved from FinRegOnt Staging via the SourceInstance property.

The import generates the composite:index. As in CFR it sorts child elements with the same parent instance. I.e. our section is the 4th child of parent chapter instance.

The composite:child part shows all elements under our section. The uslm:num value is the human readable “number” of the element, § 80b–3. The heading is a short name/text for the element, “Registration of investment advisers.

Next children are all subsections. (CFR doesn’t have subsections). We expand subsection “b”. Just like sections, subsections have type, id, identifier, style and composite index.

The composite:child part shows num value and heading. The uslm:chapeau is an introductory text for the following sub-elements, the paragraphs.

The Subsection uslm:paragraph contain the text of Private Fund exemptions to the registration requirement.
We map the staging classes from USC_USLM_Schema.ttl to the target classes in United_States_Code.ttl. The target ontology imports US_Legal_Reference.ttl, Legal_Reference.ttl and lkif-extended.ttl. The imported LKIF, Legal Reference and US Legal Reference files are the same for law and regulations.

We create a target FRO instance ontology file for each staging file.

The design challenge for the FRO USC target ontology lies in the complexity of the USLM structure. For example:

What elements can have a USLM Paragraph?
Where do Clause and Sub clause appear?
What elements are composite child of Section?
Analyzing the populated USC structure.

To answer questions about the USC structure we query and pivot population of classes and the composite:child relationship.

The SPARQL SELECT returns Parent, Child class and the number of instances.

The first line in the WHERE clause populates all 10,261 instances of the composite:child object property into ?parent and ?child variables.

The next two lines navigate to the ?parent_class and ?child_class. The last line in the WHERE clause limits the result-set to subclasses of our USLM collection class, uslm:uscDoc. We GROUP the results BY parent and child class.

We run the query against the Investor Adviser Act staging ontology, USC-2015-title15-chapter2D.ttl

The result set show:

- 3 chapeaus have a note
- 10 chapeaus have a date element.
- 45 chapeaus have a reference

This analysis drives how we design the FRO class for chapeau. For instance, we can put an OWL restriction on chapeau and the parent classes. The actual population will also drive the validation query.

The next slide shows an MS Excel pivot of the complete result set. The top row shows the USLM parent classes. The left column has the child classes. Cells contain the number of relationship per pair.
### USC USLM Title 15, Chapter 2D element pivot

#### Sum of total

<table>
<thead>
<tr>
<th>Child</th>
<th>Parent</th>
<th>chapeau</th>
<th>chapter</th>
<th>clause</th>
<th>column</th>
<th>content</th>
<th>continuation</th>
<th>header</th>
<th>heading</th>
<th>inline</th>
<th>item</th>
<th>layout</th>
<th>main</th>
<th>meta</th>
<th>notes</th>
<th>p</th>
<th>paragraph</th>
<th>proviso</th>
<th>quotedContent</th>
<th>section</th>
<th>sourceCredit</th>
<th>subchapter</th>
<th>subclause</th>
<th>subparagraph</th>
<th>subsection</th>
<th>title</th>
<th>toc</th>
<th>tocitem</th>
<th>uscDoc</th>
<th>Grand Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>34</td>
<td>59</td>
<td>3</td>
<td>1</td>
<td>42</td>
<td>113</td>
<td>209</td>
<td>1</td>
<td>126</td>
<td>597</td>
<td>603</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2016</td>
</tr>
</tbody>
</table>

#### Grand Total

|                     |        | 62      | 280     | 367    | 540    | 14      | 6          | 10      | 5      | 8      | 254  | 1      | 2      | 995  | 289  | 1195   | 1155      | 2      | 56      | 696    | 878    | 97      | 82      | 709    | 1351    | 5      | 2      | 597    | 2      | 10261     |
Extending LKIF Norm-Statute for USC USLM

Jumping ahead, the class and property browser show the outcome of our LKIF extensions.

The next pages will explain the design and population. Refer to the documentation for definitions of classes and properties:

http://finregont.com/ontology-documentation/

Financial Regulation Ontology classes have the prefix fro-usc.

The United States Code is a LKIF Statute. Both Statute and Regulation are LKIF Legal Documents* rolling up to Medium. The fro-usc:USC_Component is collection class for Level and Text entities.

- 11 USC_Level classes establish the hierarchy via the fro-leg-ref:divides object property.
- 4 USC_Text Element classes extend or annotate components via the fro-leg-ref:refers_to property, a subPropertyOf of lkif-meroo:part_of.

USLM Schema and User Guide available on the OLRC website (http://uscode.house.gov/download/download.shtml) and the pivot table is in the FinRegOnt query directory. They define the requirements for the design. For each USLM element we considered:

I. Major elements have a semantic importance indicated in the USLM documentation or have sufficient number of instances. They become USC Components. The distinction between Level and Text is structural in the XML Source.
   a) USC Level elements have an Identifier, the human readable index.
   b) USC Text Elements do not have an ID.

II. Minor elements have population of < than 10 instances.
   a) Denormalized into data and/or object properties. Examples are the Notes collection and Content class.
   b) Out of scope for FRO. We do not need table of content and formatting attributes. We skipped some elements with low population and unclear USLM definitions.

* Europe’s Alternative Investment Managers Directive (AIFMD) is a lkif-norm:Directive
The United States Code hierarchy is more complicated than the Code of Federal Regulations. The graph shows the USC levels from Title to Subsection.

USC Subchapter divides the Chapter, which in turn divides the Title. However, both Chapter and Subchapter can have Sections. Therefore the class restriction on USC_Section states OR.

The Restriction refers to a UNION (or) of the classes.

The UNION is a list of entities “[]”, referencing to Chapter and Subchapter. We read the graph as follows: Section divides Chapters or Subchapters.

The Section can be broken down further into Subsections. The Subsection divides the Section (nothing else here).
The lower elements in the USC hierarchy follow the same pattern.

Both Section and Subsection can have Paragraphs.

The Paragraph divides a Section or a Subsection.

Paragraphs can be further broken down into Subparagraphs.

The Subparagraph divides a Paragraph.
The lowest level hierarchy elements are Clause, Subclause and Item.

The Clause is a text, usually preceded with a lower case roman numeral. Subsection and Subparagraph can contain clauses.

The Subclause divides a Clause or a Paragraph.

The Item is a text fragment in a list, such as a numbered or bullet point list. Both Clause and Subclause may contain Items.

This completes the USC hierarchy in the Financial Regulation Ontology. The FRO USC Text Elements, Chapeau, Continuation, Note and Quoted Content are not dividing the hierarchy. They are text elements that annotate any type of USC Level element. FRO does not define class restriction on them.

We continue with transformations to populate our target ontology.
Mapping the USLM schema to FRO-USC

The steps defining the rules to transform USLM instance data and load into USC are similar to CFR mapping.

1. Create a new RDF/SPIN mapping file USC_15_2DspinFRO.ttl
   Import the source and target ontologies: USC-2015-title15-chapter2D.ttl and FRO_USC_Tile_12_Chapter_2D.ttl
2. Define a mapping context for each of the 15 FRO target classes in the Mapping Editor.
3. Connect data properties for simple 1:1 population.
4. Create SPIN-rules for complex data and object property transformations
5. Validate the results with introspection and queries.

To map the Paragraph we pull source and target class into the mapping editor.
Then we connect the classes to invoke the Context dialog.
For CFR our context was change namespace, but for USC we want to build the URI from the USLM ID.
Create the class mapping context

The dialog shows the selected mapping function, argument, template, a preview of results and the SPARQL expression.

The Preview Results list box shows the target URIs - a concatenation of the namespace, prefix fro-usc and the value of the USLM ID data property.

The Target function, buildURI takes one argument. The other buildURI functions operate with multiple arguments, as in a composed key.

We use the default template, fro-usc(?)1. The variable, “?1” is a placeholder for the first argument.

The SPARQL expression returns the target IRI.
• The first BIND assigns the value based ?source, an instance of uslm:Paragraph and the name of the data property.
• The Second BIND calls the SPIN function to constructs a URI based on template and argument.
USC USLM mapping context Graph

The mapping context is stored as an instance of spinmap:Context. The mapping diagram is the graph of context instances with their source and target class. (We only display 5 of the 15 contexts here).

We locate spinmap:Context in the class browser and click on the instance tab.

We select an instance and choose the Graph tab in the upper window. This will populate the graph with an instance. For example: usc-15-2D-spin:paragraph-USC_Paragraph.

We expand the graph for source class and target class. This will populate ulsm:paragraph and fro-usc:USC_Paragraph.

We can expand the class for rdfs:subClassOf to display ulsm:uscDoc and fro-usc:USC_Level.
USC USLM mapping context query

Understanding the graph, we formulate a simple SPARQL to display the mapping information.

```
SELECT ?source_class ?context ?target_class
WHERE {
  ?context a spinmap:Context ;
  spinmap:sourceClass ?source_class ;
  spinmap:targetClass ?target_class .
}
```

The query selects all spinmap:Context with their source and target classes.

Everything is within the ontology. Everything is a triple.
Transformations for USC Paragraph

We run the inference engine and take a look at the populated instance for the Investment Adviser exemption paragraph.

This is the record populated from the § 80b–3 XML we examined earlier.

The USC_Paragraph class has 6 data properties:

- **fro-leg-ref:SequenceNumber**: 3
  Direct copy of the source value in composite:index

- **fro-usc:hasId**: idd04f5b61-0c74-11e6-aa53-e455a13f2ad9
  Direct copy from uslm:id-paragraph

- **fro-usc:hasIdentifierText**: /us/usc/t15/s80b–3/b/1
  Direct copy from uslm:identifier-paragraph

- **fro-usc:hasNumberText**: (1)
  Custom Spin function to denormalize the uslm:num class instance into a data property

- **fro-leg-ref:hasComponentText**: “any investment adviser, other than …”
  This is not an ETL rule. Fro-usc:hasContentText is rdfs:subPropertyOf this data property. Population is an automatic inference.

- **fro-usc:hasContentText**: “any investment adviser, other than …”
  Custom SPIN-rule to denormalize the uslm:content class instance into a data property.

Object Properties:

- **fro-leg-ref:hasSourceInstance**: “<uslm:paragraph style="uslm-lc:I12" identifier="/us/usc/t15/s80b–3/b/1" id="idd04f5b61-0c74-11e6-aa53-e455a13f2ad9" ...>
  Custom SPIN function to set the source instance to ?this. (see CFR SPIN rule)

- **for-leg-ref:divides**: idd04f5b60-0c74-11e6-aa53-e455a13f2ad9
  The URI of the section that hold the Paragraph. A custom SPIN rule.
Using SPARQL templates for common transformations

For data properties common across USC FRO classes, we use a template rule rather than duplicating the SPARQL code.

All USC_Component instances have a source instance and heading. All USC_Level instances have a data property for the number in the index. The SPARQL CONSTRUCTs only vary by the name of the USLM source class. We can pass the class name as an argument to a SPIN template.

The snippet shows the rules, SetHeading, SetNumberText and SetSourceInstance for uslm:paragraph in the class form. We take a close look at SetNumberText.

The template has a single Argument spl:predicate, an rdf:Property. At the bottom of the rule we see uslm:id-paragraph passed to the spl:predicate.

At execution the template statement line

```
?this ?predicate ?SourceId .
```

Will be replaced with the passed argument:

```
?this uslm:id-paragraph ?SourceId .
```

And we can construct the ?targetInstance with the USC identifier.
Paragraph transformations content text

The USLM Paragraph is a generic structure. A query for the paragraph text sometimes involves several nested instances and variation.

FinRegOnt denormalizes into straight forward data and object properties, where appropriate.

The custom SPIN rule is attached to the source class uslm:paragraph. It will be executed for every instance of the class. (see CFR SPIN rules populating properties).

The CONSTRUCT shows the triple. The target (fro-usc) paragraph content text property will be populated with a “full text”.

The WHERE clause starts joining (any) child of the paragraph into a variable.

A challenge with the USLM paragraph structure is that content text is either a) directly under uslm:content b) nested within a uslm:P structure.

The OPTIONAL segments explore both possibilities and call a custom SPIN function to concatenate the text. We end up with either ?content_full_text or ?p_full_text bound.

The FILTER statements makes sure that ?full_text as a value. (some paragraphs don’t have a text.

Finally, we bind ?targetPara to the ?this variable. The spinmap:targetResource function uses the paragraph mapping context.
Concatenating complex text structures

The previous Paragraph transformation rule called a custom function to build the text.

The function usc-15-2D-spin:getUSLMComplexTypeText takes a uslm:content or uslm:p instance as an argument and returns the text.

USLM contains both content of the law and formatting. The XML reflect this having a text block broken down into small fragments of text nodes, references, dates, iterations and inline elements.

Financial Regulation Ontology is only interested in the semantic. That is human readable smallest fragment of the law, that we connect to a Legal Reasoning rule. Hence, FinRegOnt concatenates the text fragments into a string data property. Other transformation rules retain the references, uslm:ref and resolve them into object property links. For example: If a reference (="/us/usc/t15/s80b–3") points to a section, FinRegOnt populates the object property fro-leg-ref:refers_to with the URI.

The function’s query has three nested selects. The innermost query selects the text of the text of the different class instances under ?complex_type.

The next SELECT layer performs a GROUP CONCAT of the result set. The outmost query FILTERs to ensure that ?full_text is bound and casts the value to xsd:string.

http://finregont.com  © Jayzed Data Models Inc. 2016  59
Rules for paragraph object properties.

The main object properties linking FRO USC instances are fro-leg-ref:divides and fro-leg-ref:refers_to. The rule pattern is to query the composite:child structure and CONSTRUCT the target.

The rule **CONSTRUCT** sets the divided by object property for the Section of the Paragraph.

The WHERE clause joins (any) Subject with a child of ?this. We test the populated variable for being a uslm:section. (Remember that paragraphs can also occur under Subsections.)

Finally, we **BIND** target section and paragraph using the mapping context.

The second example follows the same pattern. We want check, if the paragraph has a chapeau and set it.

The CONSTRUCT sets the refers to Chapeau object property for the paragraph. (if it has a chapeau).

The WHERE clause joins composite:child of the paragraph, ?this. We test, if the child is a uslm:chapeau and BIND target chapeau and paragraph via the mapping context.

```sparql
# set paragraph divides (Section)
CONSTRUCT {
  ?targetSection fro-leg-ref:divided_by ?targetPara .
}
WHERE {
  ?section composite:child ?this .
  ?section a uslm:section .
}

# set reference to Chapeau
CONSTRUCT {
}
WHERE {
  ?this composite:child ?chapeau .
  ?chapeau a uslm:chapeau .
}
```
USC Inferencing and validation

Running the inference engine to populates FRO USC following the same steps as explained for CFR inferencing.

The screenshot shows the class browser with number of instances and the list of inference triples, scrolled for our paragraph.

There are 1985 USC_Component instances.

Initial validation follows the same steps as for FRO CFR:
1. Compare instance counts to the USCUSLM data source ontology.
2. Examine sample class instances in the Resource Form.
3. Draw and explore the graph for the Private Fund Exception.
   - Follow the hierarchy up to section and chapter.
   - Expand chapeau and notes associated with the level elements.

As for CFR we make the triples persistent in the target ontology file. The file is available on the FinRegOnt website: FRO_USC_Title_15_Chapter_2D.ttl

We won’t repeat these steps here, but rather drill deeper into data and metadata queries on the United States Code.
United States Code: “everything query”

The query SELECTs all USC instances and properties. This is to validate target population comprehensively.

The SPARQL statement is quite long (115 lines) and execution may take a few minutes. The screenshot is a zoom-out of the Excel result set (70 columns and 160,000 rows). We use the zoom-out to eyeball consistency. Any blank row should be investigated:

a) There is a break in the population or the query
b) There is a valid reason that the particular USC chapter doesn’t have the particular field(s).

The website query directory contains various queries and result sets for CFR and USC chapters in Excel format. [http://finregont.com/fro/query/](http://finregont.com/fro/query/)

The following pages show sections of the “everything query” as standalone SELECTS and results sets.
USC query: Title to Subchapter

The query SELECTs all USC instances and properties. We show the SPARQL and samples of the result set.

The WHERE clause for title and chapter is straightforward. The OPTIONAL ensures that ?subchapter is bound, dividing either chapter or subchapter.

```sparql
# USC query header information Title to Subchapter
?chapter ?chapter_seq ?chapter_ident ?chapter_heading
?chapter_number
?subchapter ?subchapter_seq ?subchapter_ident
?subchapter_heading ?subchapter_number

WHERE {
  # Title properties
  ?title a fro:usc:USC_Title ;
  fro-leg:ref:hasSequenceNumber ?title_seq ;
  fro-usc:hasIdentifierText ?title_ident ;
  fro-usc:hasHeading ?title_heading ;
  fro-usc:hasNumberText ?title_number .

  # Chapter properties
  ?chapter fro-leg:ref:hasSequenceNumber ?chapter_seq ;
  fro-usc:hasIdentifierText ?chapter_ident ;
  fro-usc:hasHeading ?chapter_heading ;
  fro-usc:hasNumberText ?chapter_number .

  # Some Titles do not have a subchapter. The Sections are directly underneath the Chapter
  # Subchapter properties
  OPTIONAL {
    ?subchapter a fro-usc:USC_Subchapter ;
    fro-leg:ref:hasSequenceNumber ?subchapter_seq ;
    fro-usc:hasIdentifierText ?subchapter_ident ;
    fro-usc:hasHeading ?subchapter_heading ;
    fro-usc:hasNumberText ?subchapter_number ;
    fro-leg:ref:divided_by ?section .
  }
}
```

<table>
<thead>
<tr>
<th>Title</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>title_seq</td>
<td>0</td>
</tr>
<tr>
<td>title_ident</td>
<td>/us/us/t15</td>
</tr>
<tr>
<td>title_heading</td>
<td>COMMERCE AND TRADE</td>
</tr>
<tr>
<td>title_number</td>
<td>Title 15—</td>
</tr>
<tr>
<td>chapter</td>
<td>4</td>
</tr>
<tr>
<td>chapter_ident</td>
<td>/us/us/t15/ch2D</td>
</tr>
<tr>
<td>chapter_heading</td>
<td>INVESTMENT COMPANIES AND ADVISERS</td>
</tr>
<tr>
<td>chapter_number</td>
<td>CHAPTER 2D—</td>
</tr>
<tr>
<td>subchapter</td>
<td>3</td>
</tr>
<tr>
<td>subchapter_ident</td>
<td>/us/us/t15/ch2D/schII</td>
</tr>
<tr>
<td>subchapter_heading</td>
<td>INVESTMENT ADVISERS</td>
</tr>
<tr>
<td>subchapter_number</td>
<td>SUBCHAPTER II—</td>
</tr>
</tbody>
</table>
We select instances of type FRO USC Subchapter where the heading matches our search criteria. And join Sections that divide the Subchapter.

The result set lists identifier, heading and section number, including the Registration of investment Advisers that we examined before.

We will continue to navigate down to the Private Fund Exemption in the next query.
Necessity of registration

Except as provided in subsection (b) and section 80b–3a of this title, it shall be unlawful...

Investment advisers who need not be registered

Any provision of this subchapter (other than subsection (a) of this section) which prohibits...

Procedure for registration; filing of application; effective date of registration; amendment of registration

The Commission, by order, shall censure or place limitations on the activities of any person...

Registration of successor to business of investment adviser

Any successor to the business of an investment adviser registered under this section shall be...

Withdrawal of registration

Any person registered under this section may, upon such terms and conditions as the Commission finds...

Money penalties in administrative proceedings

In any proceeding in which the Commission may impose a penalty under this section, ...

Authority to enter order requiring accounting and disgorgement

In any proceeding in which the Commission may impose a penalty under this section, ...

Exemption of and reporting by certain private fund advisers

In prescribing regulations to carry out the requirements of this section with respect to investment advisers acting as...
### USC query: Registration Subsection

The query SELECTs all Paragraphs under Subsection § 80b–3/b - Registration Exemption

Finally, we drilled down from Title 15 all the way to the Investment Adviser Registration exemption. The piece of USC OLCR XML that we started with.

Note that paragraphs 5,7 and don’t have a text, because they have content in Subparagraphs.

<table>
<thead>
<tr>
<th>subsection_para_ident</th>
<th>subsection_para_number</th>
<th>subsection_para_text</th>
</tr>
</thead>
<tbody>
<tr>
<td>/us/usc/t15/s80b-3/b/1</td>
<td>(1)</td>
<td>any investment adviser, other than an investment adviser who acts as an investment adviser to any private fund, all of whose clients are residents of the State within which such investment adviser maintains his or its principal office and place of business, and who does not furnish advice or issue analyses or reports with respect to securities listed or admitted to unlisted trading privileges on any national securities exchange;</td>
</tr>
<tr>
<td>/us/usc/t15/s80b-3/b/2</td>
<td>(2)</td>
<td>any investment adviser whose only clients are insurance companies;</td>
</tr>
<tr>
<td>/us/usc/t15/s80b-3/b/3</td>
<td>(4)</td>
<td>any investment adviser that is a foreign private adviser;</td>
</tr>
<tr>
<td>/us/usc/t15/s80b-3/b/4</td>
<td>(5)</td>
<td>any plan described in section 414(e) of title 26 , any person or entity eligible to establish and maintain such a plan under title 26 , or any trustee, director, officer, or employee of or volunteer for any such plan or person, if such person or entity, acting in such capacity, provides investment advice exclusively to, or with respect to, any plan, person, or entity or any company, account, or fund that is excluded from the definition of an investment company under section 80a–3(c)(14) of this title;</td>
</tr>
<tr>
<td>/us/usc/t15/s80b-3/b/5</td>
<td>(6)</td>
<td></td>
</tr>
<tr>
<td>/us/usc/t15/s80b-3/b/6</td>
<td>(7)</td>
<td></td>
</tr>
<tr>
<td>/us/usc/t15/s80b-3/b/7</td>
<td>(8)</td>
<td></td>
</tr>
</tbody>
</table>
Summary and conclusion


Government publishers provide the laws and regulations in XML format.
- **Extract and convert the source file into FRO RDF Staging.**
- **Transform the RDF representation into FRO ontology with semantic mapping and Inference Rules.**
- **Load the inference triples into the target FRO ontology.**

The result is a standard Legal Ontology (LKIF) with FRO extensions populated with the full text of Finance Laws and Regulations. The Semantic Web approach has everything within the ontology and available for SPARQL query: Requirements, Schema, Data, Linage to source, and Mapping.
Chapter II – books, recommended companion reading

Data Integration Blueprint and Modeling
Anthony David Giordano
IBM Press, 2011

Law and the Semantic Web: Legal Ontologies, Methodologies, Legal Information Retrieval, and Applications
Richard Benjamins, Pompeu Casanovas, Joost Breuker, Aldo Gangemi
Springer, 2009

Model Driven Engineering and Ontology Development
Dragan Gasevic, Dragan Djuric, Vladan Devedzic
Springer, 2010

Legal Ontology Engineering
Nuria Casellas
Springer, 2011
Chapter II - references

1. Financial Regulation Ontology
   iii. SPARQL queries and result sets in Excel: http://finregont.com/fro/query/


6. SPIN – SPARQL Inferencing Notation website: http://spinrdf.org/

7. SPARQL Motion website: http://sparqlmotion.org/
Chapter 3 will be added to FinRegOnt.com as a separate deck in November.