RESEARCH ARTICLE

Population of a spatio-temporal knowledge base for jurisdictional domains

Javier Lacasta[a*], Francisco Javier Lopez-Pellicer[a], Aneta Florczyk[b],
Francisco Javier Zarazaga-Soria[a] and Javier Nogueras-Iso[a]

[a]Computer Science and Systems Engineering Department,
Universidad de Zaragoza, María de Luna 1, 50018, Zaragoza, Spain.

[b]European Commision Joint Research Centre,
Enrico Fermi 2749, 21027, Ispra Varese, Italy.

(*Received 00 Month 200x; final version received 00 Month 200x*)

Jurisdictional domains are legal divisions of the Earth's surface frequently used in classification and search systems for location-based queries. However, there are few compilations of jurisdictional domains that include their evolution. One of the causes is the complexity of their generation. As an advance in this area, this paper describes a process that facilitates the creation of knowledge bases containing the evolution of jurisdictional domains. As application example, this paper presents the creation of a jurisdictional knowledge base of Spain with its evolution between 1830 and 2011.

Keywords: Ontology population, Jurisdictional domain, Administrative model, Spatial representation, Temporal management

1. Introduction

Jurisdictional domains are *de facto* or formally recognized entities that cover specific areas of the Earth's surface. They are units of administration for local, regional, national or international governance, and evolve over time as political actions change their responsibilities, extension, number, or type. This broad concept comprehends a wide range of entities including new (e.g. South Sudan), extinct (e.g. Yugoslavia) and disputed (e.g. Palestine) states and local governments.

These entities are commonly used for classification and search as they provide a widespread way to identify earth regions. However, these entities are not static; they

[*] Corresponding author. Email: jlacasta@unizar.es

ISSN: 1365-8816 print/ISSN 1362-3087 online

DOI: 10.1080/1365881YYxxxxxxxx
http://www.informaworld.com

change along time due to political factors. This creates the need of using different jurisdictional models for different periods of time. Not only that, the equivalences between these models may not be obvious as changes may deeply affect to the hierarchical divisions of a country (Özçep *et al.* 2012).

This continuous evolution complicates the use of jurisdictional domains to search in collections containing historical information. An example of this problem can be observed in the Old Maps Online project (Pridal and Zabicka 2008). In this system, the access to historical maps using place names is performed using current jurisdictional divisions. This requires the users to know the historic and current organization of the desired areas as they need to search historic places using current names. For example, a search about the old kingdom of Prussia (1525-1947) requires a query that includes current Germany, Poland, Russia, Lithuania, Denmark, Belgium, Czech Republic and Switzerland. Additionally, since the union of these countries includes areas that were not part of Prussia, it is needed to filter out the unrelated results. Figure 1 shows the boundary changes in Prussia between 1807-1871 to illustrate how fast the jurisdictional boundaries of a region can change, and how numerous these changes can be.

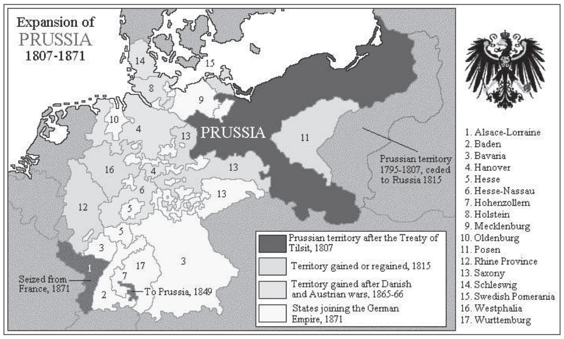


Figure 1. Boundary changes in Prussia between 1807-1871 (Carr 2006)

In the literature, there are some approaches to model the evolution of jurisdictional domains that can be used in this context (Kauppinen and Hyvönen 2007, López-Pellicer *et al.* 2011, Gantner *et al.* 2012). However, the population of such models (i.e., addition of instances) with the evolution of jurisdictional domains over a long period is not an easy task. The required information may be distributed along a heterogeneous collection of documents, and it has to be cleaned and integrated. Not only that, the types of jurisdictional domains and divisions are different between countries and they are available in different ways (in format and structure). This makes that the software designed to populate the evolution model of a country is not usually valid for other one.

This paper proposes a semi-automatic process that has as objective to reduce the complexity of creating and populating these kind of models. It does so by dividing the generation process in a collection of successive tasks, many of which are independent of the specific details of the data used to populate the model. The reuse of the software components implementing these independent tasks allows reducing the generation work to: (1) the selection of the sources to process, (2) the definition of the structure of the model to generate, (3) the transformation of the source content into the format accepted by the reusable components, and (4) the correction of source inconsistencies.

This process is designed to generate the jurisdictional domains and their evolution as a knowledge base. That is, an information repository used for knowledge management consisting of an ontology describing a conceptual model (Guarino 1998) and the instances of such a model. The knowledge bases have already been successfully used in the Semantic

Web to represent jurisdictional domain evolution (see Section 4) and provide a more direct way of modeling the administrative, spatial and temporal features of jurisdictional information than classical relational models (Murray 2006, Cotelo Lema 2006) (e.g., the transitive closures in jurisdictional *part-of* structures, and the linear evolution of temporal changes).

The proposed process has been used to create a knowledge base containing the evolution of Spanish jurisdictional domains between 1830 and 2011 (former overseas territories are not included). The paper describes the data sources, the transformations performed, the issues corrected, and the structure of the resulting knowledge base.

The paper is organized as follows. Section 2 details the proposal to generate and populate a knowledge base with jurisdictional information from heterogeneous sources. Section 3 shows the application of the process to create the Spanish model. Section 4 describes the related work and compares it with our approach. The paper finishes in Section 6 with some conclusions, and an outlook on future work.

2. Knowledge Base creation

The process to generate the desired jurisdictional knowledge base consists of the three phases shown in Figure 2. First, it is required to select the sources to process (*Compilation*). They may include any document (web, databases, books, . . .) describing the jurisdictional structure of the selected country in the desired time-span. They have to provide the types of jurisdictional domains, the instances of these types, and the changes they have suffered. Then, the ontology used as schema of the knowledge base has to be defined (*Country ontology definition*). Our process uses the ontology defined in López-Pellicer *et al.* (2011) as base of this schema. The work to perform in this step is the extension of this ontology with the types of jurisdictional domains discovered in the sources. The last step processes the source content and transforms it into instances of the ontology defined as schema (*KB population*).

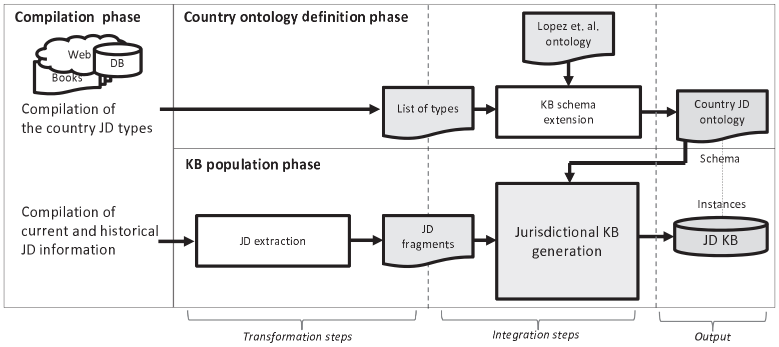


Figure 2. Process to generate the jurisdictional knowledge base of a country

These phases are common to other knowledge generation approaches in the Semantic Web (Villazón-Terrazas *et al.* 2011, Lacasta *et al.* 2013, 2010), and in the jurisdictional information field (see Section 4). However, this process differs in the way that information is processed. Instead of a monolithic approach to process the sources, we opted by a modular one. This approach allows a fine-grained configuration of the knowledge base schema and the population workflow. It also reduces the amount of new software required to generate a knowledge base with different requirements (e.g., a different country).

For example, this loosely coupled approach avoids the need to adapt the integration algorithms every time a new set of sources is used.

The rest of the section describes each phase of the process, their features, and limitations. Section 2.1 describes the types of jurisdictional sources and the kind of information they have to provide to be usable in the process. Section 2.2 describes how to extend the knowledge base schema. Section 2.3 describes the population of the knowledge base. Specifically, it shows the format required as input of the process and the integration algorithms developed to generate the jurisdictional knowledge base.

2.1. *Compilation phase*

The generation of a jurisdictional knowledge base requires knowing the complete structure of the jurisdictional domains in the selected time span (types, instances, properties and evolution along time). The main sources of this information are data collections published by official institutions of the analyzed country.

The list of jurisdictional domain types can be obtained from official documents describing the administrative organization, the analysis of jurisdictional data collections, and the consultation with experts in the legislative field.

With respect to instances, properties and evolution, it is needed to rely on databases or repositories created for statistical purposes. One of the ways this information can be found is as a set of snapshots. Each snapshot describes the jurisdictional domains in a time span without jurisdictional changes. To be able to reconstruct the evolution of the jurisdictional domains, the relations between consecutive snapshots have to be explicit. An example of this approach is the SALB project (Ebener *et al.* 2004) created by the United Nations to provide the evolution of world first and second administrative divisions.[1] The alternative way this information is found is as a collection of jurisdictional changes. These changes can be put into context using as reference a single snapshot at a date in the period covered by the changes. This is the approach used in collections such as the Code History Database of the UK Office for National Statistics,[2] the Registry of Local Entities of Spain,[3] or the country codes from ISO-3166.[4]

In our process, the input must be composed of a list with the jurisdictional changes in the desired period, and a snapshot of the state of the jurisdictional domains just after the last change (see Section 2.3.1). One of the tasks to perform is the creation of the transformation tools that convert the sources to the input format. Therefore, in addition to the quality, a factor for selecting a source is the simplicity to perform this transformation. The decision of using a change based input has technical roots. We find technically easier the conversion of a snapshot collection into a list of changes than vice versa.

2.2. *Country ontology definition phase*

Once the sources have been selected, the next step is the definition of the ontology used as schema of the knowledge base. For this task, we decided to use the jurisdictional domain

[1] http://www.ungiwg.org/coreDB

[2] http://www.ons.gov.uk/ons/guide-method/geography/products/names--codes-and-look-ups/code-history-database/index.html

[3] http://www.seap.minhap.gob.es/dms/es/publicaciones/centro_de_publicaciones_de_la_sgt/Monografias0/parrafo/011113/text_es_files/Variaciones-INTERNET.pdf

[4] http://www.iso.org/iso/home/standards/country_codes/updates_on_iso_3166.htm

ontology (JDO) described in López-Pellicer *et al.* (2011) extended with the types of jurisdictional domains of the country to model.[5] The ontology determines the structure of the result, and therefore the functionality the generation components have to provide. The use of a different one is possible, but it would require replacing the generation components described in Section 2.3 with other ones able to deal with the new schema.

JDO models the administrative, spatial and temporal features of jurisdictional information. It contains the *administrative-division* class which represents any type of jurisdictional domain division. Its purpose is to be specialized with the types of jurisdictional domains of the modeled countries (and their relations). For example, to model the administrative units of Spain, the types of jurisdictional domains such as "Comunidad Autónoma" (Autonomous Community) and "Provincia" (Province) are defined as specialization of the administrative division class. The changes are represented through the *duration* object-property. It identifies the time-interval (with an *initial-instant* and a *final-instant*) in which each concept, property, or relation is officially recognized. For example, duration can be used for describing the period on which a name was used as the official name of a jurisdictional domain. The addition of the duration to properties and relations is performed through reification. Additionally, the evolution of jurisdictional domains is modeled by the *successor* property. It is specialized with the different types of succession relations required. Figure 3 shows the successor relations that can be currently managed by the process components. For example, to indicate that a jurisdictional domain (such as a municipality) is absorbed by another one, we would use the incorporation relationship. They have been selected as they cover the most common types of jurisdictional changes, and they can be combined to build other relations (e.g. the barter of territories is composed of consecutive although simultaneous partial incorporations). It is possible to explicitly add a new type of successor to the process, but it requires the creation of a new module in the generation phase that processes the new type of relation.

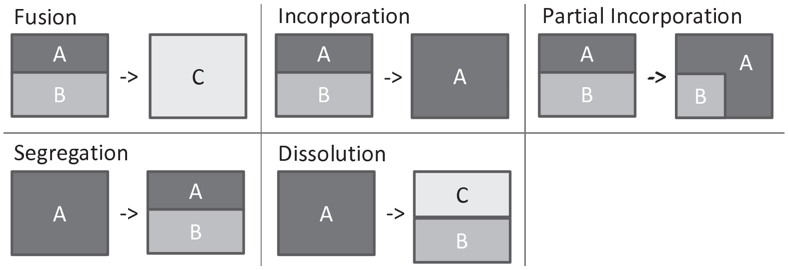


Figure 3. Types of succession relations

The extension of the JDO consists of adding the types of jurisdictional domains obtained in the compilation phase as subclasses of the JDO *administrative-division* class. Each new class should bear its official name, a reference to the country they belong to and the description of allowed *part-of* relations. Our experience is that the number of different types of jurisdictional domains is small even considering a large period of time as they quite stable. For example, there are only five main types of jurisdictional domains in Spain (see Section 3.1) and they have not changed since 1931. This reduced number makes easy the addition to the ontology using any ontology editing tool such as Protégé.[1]

[5] `https://raw.github.com/stjdo/stjdo/master/JurisdictionalDomainSchema.owl`

[1] `http://protege.stanford.edu`

The main drawback of the selected approach is the structural limitation of reification in terms of reasoning and model complexity. On the one hand, there is a lack of reification support in description logic based reasoners. This is not a great issue as complete automatic reasoning is not vital in our context, and if required, there are solutions to support it (Zimmermannc *et al.* 2012). However, it can be a problem in contexts using reasoners that do not implement such solutions. On the other hand, reification produces a significant increase in the size of RDF models (each reified property adds 5 RDF triples). This should not be an issue with top storage systems,[2] but it may limit the number of systems able to deal with it. There are other jurisdictional ontologies in the literature that could have been used as base of the knowledge base schema (see Section 4). However, the way they manage jurisdictional evolution also has its issues. The main approaches model spatio-temporal evolution as a succession of resources, each one describing the complete state of a jurisdictional domain between two changes (of any property), and provide the mechanism to relate those that are part of a succession chain (Kauppinen and Hyvönen 2007, Gantner *et al.* 2012). In jurisdictional models that change frequently this may lead to a proliferation of instances because the number of generated instances is directly proportional to the total number of property changes. We prefer JDO because it describes the changes at property level (each property can have its own time span) using reification. On the one hand, this avoids the need to create a new instance for the same jurisdictional domain after each change. Only the properties that change are added. On the other hand, it makes easy to add additional types and temporal properties (e.g., legal attribution of each type of jurisdictional domain, name of mayors of a municipality. . .). It does not require splitting existing resources due to a value change of the new property. Additionally, the separation of temporal information from the core structure of the model makes possible the inclusion of properties with unknown of fuzzy temporal references (e.g., a name that we do not know the exact dates in which it was official). Another existent alternative is the use of named graphs to model periods of time (Tappolet and Bernstein 2009). In this work, each required period of time is defined as a different named graph. Each property is added to the graph that describes the existence period of the property. This allows a compact representation of the information, but it requires a SPARQL (W3C 2013) extension to allow temporal queries to the graphs. Due to this, we prefer the use of JDO, as it can be queried using existent SPARQL tools.

2.3. *Knowledge base population phase*

The data sources to use in the process may be very heterogeneous in their formats (plain text, tabular structures, html, databases, . . .) and organization of the information (Hart and Dolbear 2013). To deal with this issue, we separate the extraction of the source content from the integration of this information in the knowledge base (see *KB Population Phase* of Figure 2). This is done by transforming all the sources into a common format and using it as input of the generation process (Section 2.3.1)

The input model represents the jurisdictional domain evolution as a collection of changes in the jurisdictional domain along the selected time-span, and a snapshot of the state of the jurisdictional domains after the last change. It is not needed that each input record contains all the information about a jurisdictional domain or a change. This information can be distributed in several input resources, each one obtained from a different source (input fragment). The integration of these fragments is done by the

[2] http://franz.com/agraph/allegrograph/

population components described latter in this section. This approach reduces the effort to generate a new jurisdictional model. The components that perform the integration do not need to be changed when a new set of sources is used as they always receive their input in the same format. Additionally, since the input does not require merged information, the generation of the input is reduced in many cases to a simple data format transformation.

Once the source content has been represented according to this input format, it is integrated by the components described in Figure 4. The *fragments merger* components integrate the input fragments to generate complete jurisdictional domains (current snapshot) and change descriptions (Section 2.3.2). The *jurisdictional domain generator* identifies the relations between the input jurisdictional domains and generates them as instances of the extended JDO (Section 2.3.3). The result of this step is a knowledge base (JD KB) with the jurisdictional domains of the input snapshot. Finally, the *jurisdictional domain evolution generator* adds the change information to the jurisdictional domains giving as result the final knowledge base (Evolution JD KB) with the evolution of jurisdictional domains in the selected time span (Section 2.3.4). This process is a kind of ontology population from textual mentions (Magnini *et al.* 2006). In this context, the set of entities to be added is clearly stated and the identification of their types and relations is done by using pattern based approaches and analyzing (when required) the context of the extracted information records.

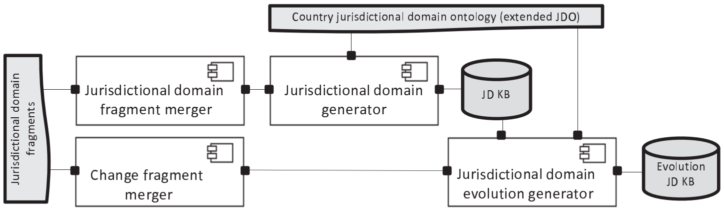

Country jurisdictional domain ontology (extended JDO)
Jurisdictional domain fragments
Jurisdictional domain fragment merger
Jurisdictional domain generator
JD KB
Change fragment merger
Jurisdictional domain evolution generator
Evolution JD KB

Figure 4. Components of the Jurisdictional KB generation step

To describe the process functionally, the following sub sections contain pseudo-algorithms using the following mathematical symbols: $\leftarrow$ (assignment), $\in$ (in), $\cup$ (union), $\wedge$ (and), $+$ (concatenate), $|$ (such that), and $\emptyset$ (empty set). Additionally, for the sake of clarity, these algorithms include the SPARQL queries used to transform and integrate the input records.

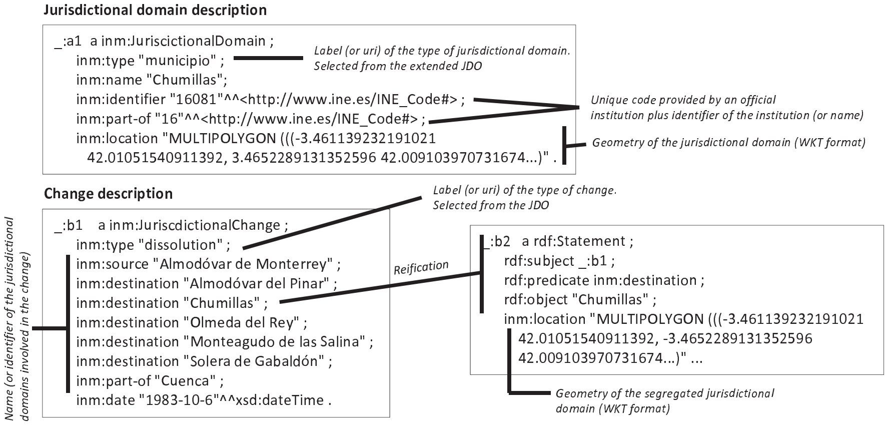
2.3.1. Population input model

The input model is inspired in the conceptual schema for gazetteers defined by ISO-19112 standard (ISO 2003) and it is extended with elements to represent temporal changes.[1] It describes the properties of the jurisdictional domains in the selected snapshot: names, types, identifiers, geometries, and *part-of* relations. To make the input more flexible, *part-of* relations may contain the name of the upper domains or their official identifiers. To generate the jurisdictional model evolution between a date and the present, this part of the input has to contain the current jurisdictional domains of the country. With respect to the jurisdictional changes, it provides properties to indicate: the type of change, the involved jurisdictional domains (name and/or identifier of source and destination jurisdictional domains), the date, and optionally, the name of upper jurisdictional

[1] `https://raw.github.com/stjdo/stjdo/master/PopulationInputModel.owl`

domains. The allowed types of changes are those indicated in Section 2.2 (*fusion*, *segregation*, *incorporation*, *dissolution*, *partial incorporation*). Geometry modifications are provided as reification of the change source/destination properties.

Figure 5 shows an example of use. Concept *_:a1* describes the municipality of *Chumillas* (Cuenca - Spain). It includes its type, name, identifier, upper jurisdictional domain (part-of) and geometry. Concepts *_:b1* and *_:b2* show the creation of *Chumillas* in 1983 as result of the dissolution of *Almodóvar de Monterrey*. It provides the type of change, the domains involved (source and destination fields), and the change date. The involved jurisdictional domains can be annotated with their geometries through reification. Since these resources are not final, blank nodes are used as identifiers. This example contains the whole description of a resource or a change. However, as indicated previously, this information can be distributed in several resources. For example, one may contain the type and name, other may have the identifier and location, and so on. The restrictions about minimum content required in a fragment to be usable in the process are described in Section 2.3.2.



Jurisdictional domain description

```
_:a1  a inm:JuriscictionalDomain ;
    inm:type "municipio" ;
    inm:name "Chumillas";
    inm:identifier "16081"^^<http://www.ine.es/INE_Code#> ;
    inm:part-of "16"^^<http://www.ine.es/INE_Code#> ;
    inm:location "MULTIPOLYGON (((-3.461139232191021
      42.01051540911392, 3.4652289131352596 42.009103970731674...)" .
```

Change description

```
_:b1    a inm:JuriscdictionalChange ;
    inm:type "dissolution" ;
    inm:source "Almodóvar de Monterrey" ;
    inm:destination "Almodóvar del Pinar" ;
    inm:destination "Chumillas" ;
    inm:destination "Olmeda del Rey" ;
    inm:destination "Monteagudo de las Salina" ;
    inm:destination "Solera de Gabaldón" ;
    inm:part-of "Cuenca" ;
    inm:date "1983-10-6"^^xsd:dateTime .
```

```
_:b2   a rdf:Statement ;
    rdf:subject _:b1 ;
    rdf:predicate inm:destination ;
    rdf:object "Chumillas" ;
    inm:location "MULTIPOLYGON (((-3.461139232191021
      42.01051540911392, -3.4652289131352596
      42.009103970731674...)" ...
```

Label (or uri) of the type of jurisdictional domain. Selected from the extended JDO
Unique code provided by an official institution plus identifier of the institution (or name)
Geometry of the jurisdictional domain (WKT format)
Label (or uri) of the type of change. Selected from the JDO
Name (or identifier of the jurisdictional domains involved in the change)
Reification
Geometry of the segregated jurisdictional domain (WKT format)

Figure 5. Representation of a jurisdictional domain and a change involving the domain according to the defined model (in RDF/Turtle)

The use of RDF as input model is a requirement derived from the implementation of Section 2.3.3 and 2.3.4 components. These components perform the transformation/integration of an RDF model (the input) into another one (the JDO). This approach has the advantage that all the components performing the transformation are homogeneous. They all make use of the Semantic Web stack to transform and interchange the data in RDF. This is useful as many of the integration operations require complex matching between elements in the model that can be directly implemented with SPARQL queries. Another advantage is the simplification of the processing of new types or properties (e.g., neighborhood relations). It only implies to extend the input, the JDO, and the creation of an additional integration procedure for these new elements. Finally, the existence of RDF visualization tools, such as GraphViz (Ellson *et al.* 2002), simplifies the manual validation of the correctness and quality of the extracted information.

The specific details of how the input can be generated from a given set of sources are outside the paper objectives. There is abundant literature on model transformation techniques that can be used for this task. In general, the use of rules adapted to the source structure, in the way described by Maynardand *et al.* (2008) or Gregory *et al.* (2011) can prove to be a good transformation approach, but other techniques can be used if needed. For structured sources such as databases, in the ontology learning context, tools like OntoStudio (Weiten 2009) or KaOn Reverse (Stojanovic *et al.* 2002) allow the establishment of mappings between heterogeneous structures and ontologies. In the Semantic Web context, tools such as ODEMaster (Rodriguez and Gómez-Pérez 2006) or D2RServer (Bizer and Cyganiak 2006) provide similar functionality. If the sources are semi-structured or unstructured, Gregory *et al.* (2011) propose a rule/context based process that allows the identification of instances. It uses a customized rule-based mapping for structured sources, and a natural language based processing that searches for patterns and co-occurrences for unstructured elements. As conversion example, Section 3 shows the detail of the transformation of the sources used to generate the jurisdictional information of Spain.

2.3.2. Fragments integration

This section describes the algorithms defined to merge the input fragments into the complete jurisdictional domains and change descriptions required to generate the jurisdictional knowledge base. To be able to automatically establish equivalences between the input fragments, these algorithms require the fragments content partially overlap in the ways indicated in the algorithm descriptions.

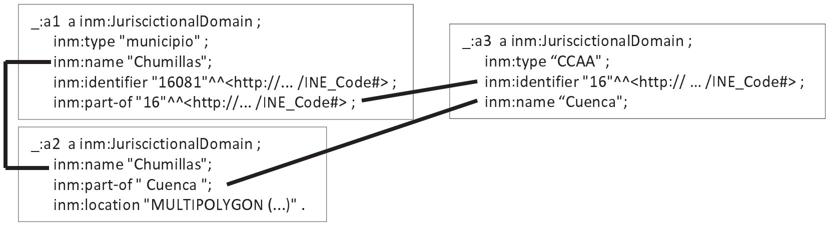
Algorithm **1** performs the integration of the jurisdictional domain fragments (*jdFrag*). It takes each fragment in the collection (*jdf*) and searches for equivalences. It assumes that the *inm:identifier* in the jurisdictional domains is unique, and the composition of *inm:name* and *inm:part-of* relations of a jurisdictional domain is also unique (first SPARQL query). The resources with the same value in these properties are merged (last part of the first *for*). If these assumptions were not valid, a complementary way to distinguish the jurisdictional domains would be needed.

A characteristic of the input is that *part-of* relations may contain the name of the upper domains or their official identifiers. The identification that a name is equivalent to an identifier in a *part-of* relation is performed by the second part of the algorithm. In this case, the equivalence is identified by locating if there is an upper jurisdictional domain that has the required identifier and name (second SPARQL query). The process has to be repeated until no more fragments are fused because of the hierarchical structure of the jurisdictional domains. Until the upper jurisdictional domains are integrated, the lower ones may not have the resource with the required identifier/name pair.

Algorithm 1: Integration of jurisdictional domain changes.

```
procedure JDFRAGMENTINTEGRATION($jdFrags$)
    for $jdf \in jdfFrags$ do                                   ▷ Identification of direct equivalences
        $relFrags \leftarrow$ SELECT ?eqf WHERE { {jdf :identifier ?i. ?eqf :identifier ?i}
        UNION {jdf :name ?n. ?eqf :name ?n. jdf :part-of ?p. ?eqf :part-of ?p}. FILTER (jdf!=?eqf)}
        $jdf \leftarrow jdf \cup relFrags$
        $remove(jdfFrags, relFrags)$
    end for
    repeat                                                      ▷ Identification of indirect part-of equivalences
        $foundEquivalence \leftarrow false$;
        for $jdf \in jdfFrags$ do
            $relFrags \leftarrow$ SELECT ?eqf WHERE {jdf :name ?n. ?eqf :name ?n.
            { {jdf :part-of ?p. ?x :identifier ?p. ?eqf :part-of ?y. ?x :name ?y } UNION
            { jdf :part-of ?p. ?x :name ?p. ?eqf :part-of ?y. ?x :identifier ?y }}. FILTER (jdf!=?eqf)}
            if $relFrags \neq \emptyset$ then
                $jdf \leftarrow jdf \cup relFrags$
                $remove(jdfFrags, relFrags)$
                $foundEquivalence \leftarrow true$;
            end if
        end for
    until $notfoundEquivalence$
    return $jdFrags$
end procedure
```

Figure 6 shows a simplified example of how the generation of the "Chumillas" jurisdictional domain in Figure 5 is done in base to a set of fragments. It can be observed how the identification of string equivalences of names, identifiers and *part-of* relations allows the merge of the information.



```
_:a1  a inm:JuriscictionalDomain ;
    inm:type "municipio" ;
    inm:name "Chumillas";
    inm:identifier "16081"^^<http://... /INE_Code#> ;
    inm:part-of "16"^^<http://... /INE_Code#> ;
```

```
_:a2  a inm:JuriscictionalDomain ;
    inm:name "Chumillas";
    inm:part-of " Cuenca ";
    inm:location "MULTIPOLYGON (...)" .
```

```
_:a3  a inm:JuriscictionalDomain ;
    inm:type “CCAA" ;
    inm:identifier "16"^^<http:// ... /INE_Code#> ;
    inm:name “Cuenca";
```

Figure 6. Example of integration of jurisdictional domain fragments

The algorithm used to perform the integration of the change fragments is very similar to Algorithm **1**. The only difference is the set of rules used to determine when two change fragments are equivalent. In this case, this happens when the fragments have the same value of *inm:type*, *inm:date* and *inm:part-of*. Additionally, depending of the type of change, the fragments must have the same *inm:source* or *inm:destination*. For example, the common element in a fusion is the destination, but in a segregation it is the source.

If any resource is left incomplete after the merge process (e.g., a jurisdictional domains without name), it is reported that there are errors in the sources that prevents the identification of equivalences for the incomplete resources. The process cannot continue if the input is not valid, so they have to be reviewed and corrected.

2.3.3. Jurisdictional domain generator

This section describes the algorithm used to transform the jurisdictional domains obtained from the *fragments integration* step (snapshot of the jurisdictional model at a given date) into instances of the JDO ontology (Algorithm **2**). It requires as input: the integrated jurisdictional domains (*jdCol*), the namespace of the instances to generate (*baseUri*), and the JDO. As result, it generates the JDO instances (jdKB). First, the final URI of each resource is generated in base to the *inm:identifier* field or the *inm:name* and *inm:part-of* fields. Second, the class to instance in the extended JDO is obtained from the *inm:type* and the ontology definition. Then, the JDO instance is created from the information of the source (first SPARQL construct).[1] Finally, the *part-of* relations between the instances are generated (second SPARQL construct) by locating the jurisdictional domain with the identifier or name contained in the *part-of* relations. This is needed as the input *part-of* references are implicit (a name, or identifier of the upper jurisdictional domain), and they have to be converted into explicit references (URIs).

Algorithm 2: Generation of current jurisdictional domain instances.

```
procedure CURRDOMGEN($jdfCol, baseUri, JDO$)
    $jdKB \leftarrow \emptyset$
    for $jd \in jdCol$ do                                ▷ Transformation into JDO instances
        if $jd.identifier = \emptyset$ then
            $jd.identifier \leftarrow jd.part\text{-}of + jd.name$
        end if
        $jdi \leftarrow \emptyset$
        $jdi.uri \leftarrow baseUri + jd.identifier$
        $jdi.type \leftarrow jdTypeUri \in JDO | ends(jdTypeUri, jd.type)$
        $jdi \leftarrow jdi \cup$ CONSTRUCT {jdi skos:prefLabel ?x ...} WHERE {jd :name ?x ...}
```

[1] The construct operation has been shortened for clarity purposes.

```
        $add(jdi, jdKB)$
    end for                                                    ▷ Addition of part-of relations
    $uriPartOf \leftarrow$ CONSTRUCT {jdi :part-of ?x} WHERE {jdi :part-of ?p. {{?x dc:identifier ?p}
    UNION {?x :name ?p }. jdi rdf:type ?t1. ?x rdf:type ?t2. FILTER(?t1!=?t2)}
    $delete(jdKB, :part\text{-}of)$
    $add(jdKB, uriPartOf)$
    return $jdKB$
end procedure
```

2.3.4. Jurisdictional domain evolution generator

This section describes the algorithms used to extend the jurisdictional instances generated in the *Current JD generator* step with evolution information provided by the collection of changes generated in the *fragments integration* step. The resulting collection of instances is the desired jurisdictional knowledge base. Algorithm **3** describes the general process for this task. Algorithm **4** focuses on the details of processing the set of changes at a date.

Algorithm **3** receives as input the collection of changes (*jdChCol*), the previously generated instances (*jdKB*), and the instance namespace (*baseUri*). It uses the changes to extend the collection of instances with historical jurisdictional domains and their changes. First, it sorts the changes in descendent order (*sort* function). Then, the changes with the most recent date are selected (changes with the same change date as the first element in *jdChangeCol*). Those changes are added to the knowledge base, in the way described later (*addChangesToKB*), to extend the covered time span. Finally, the processed changes are deleted and the process is repeated until all the changes are added to the knowledge base (*evolutionJDKB*).

Algorithm 3: Generation of historical jurisdictional domain instances.

```
procedure CURRDOMGEN($jdChangeCol, jdKB, baseUri$)
    $sort(jdChangeCol)$                                        ▷ Sort by date in descendent order
    while $jdChangeCol \neq \emptyset$ do
        $recentChanges \leftarrow \emptyset$
        $add(recentChanges, change) | change \in jdChangeCol \land change.date = jdChangeCol[0].date$
        $evolutionJDKB \leftarrow addChangesToKB(recentChanges, jdKB, baseUri)$
        $delete(jdChangeCol, recentChanges)$
    end while
    return $evolutionJDKB$
end procedure
```

Each set of changes that occurred at a specific date are added to the knowledge base using Algorithm **4**. It receives as input a set of changes (*changes*), and the knowledge base to enrich (*jdKB*), and it extends the instances with the information in the changes. This algorithm relies on the high quality of the previously generated jurisdictional model. If it is incomplete, erroneous, or it does not correspond to the state of the jurisdictional domains after the last change, it will not be possible to identify the parts of the jurisdictional model the change is making reference to. This assumption is not applicable the list of changes as it can come from old legislations or cadasters that may contain imprecisions or errors. Therefore, the algorithm takes measures to identify and correct possible inconsistences when adding the changes to the jurisdictional knowledge base.

Algorithm 4: Addition of changes to the knowledge base.

```
procedure ADDCHANGESTOKB($changes, jdKB, baseUri$)
    $size \leftarrow 0$
    while $changes \neq \emptyset \land changes.size \neq size$ do          ▷ Add correct changes
        $size \leftarrow changes.size$
        for $change \in changes$ do
            $jurDomInChange \leftarrow getJDInChange(change, jdKB)$
            if $isComplete(change, jurDomInChange)$ then
                $applyChange(change, jurDomInChange, jdKB, baseUri)$
                $remove(change, changes)$
            end if
        end for
    end while
    for $change \in changes$ do                ▷ Add changes with jurisdictional domain name errors
        $map(change, jdKB) \leftarrow getLevJDInChange(change, jdKB)$
        $reviewMatches(map(change, jdKB))$
        $applyChange(change, jdKB, baseUri)$
        $remove(change, changes)$
    end for
    return $jdKB$
end procedure
```

The first step to add a change is to search the jurisdictional domains contained in the change description in the knowledge base (*getJDInChange*). This is done by comparing the names and *part-of* relations in the same way as in Algorithm **2** (String comparison of name and part-of relation values). Polysemy issues (names reused over time) are managed by analyzing the official recognition periods of each jurisdictional domain that are spatially and temporally compatible. That is, the correct jurisdictional domain to match a given name has to exist at the change date (temporal compatibility) and has to be located in the same region than the rest of the jurisdictional domains involved in the change (spatial compatibility).

If all the jurisdictional domains that exist after the change are matched (*isComplete*), the change is applied (*applyChange*). Each type of change (*Denomination change*, *Partial incorporation*, *Segregation*, *Incorporation*, *Fusion*, and *Dissolution*) requires a different set of updates. Because the knowledge base is generated in inverse temporal order, each change generates an action that is inverse to its meaning (e.g. a *Dissolution* produces the addition to the knowledge base of the dissolved jurisdictional domain).

Integrating a *denomination change* implies the addition of the name of the jurisdictional domain previous to the change date. Additionally, the values of the *final_instant* of the added historical name, and the *initial_instant* of the name that replaces it, have to be set to the change date. The rest of the types of changes require the addition of a new relation of the corresponding type between the involved domains. Then, the values of the *initial-instant* of the created jurisdictional domains and the *final-instant* of the extinct jurisdictional domains have to be set to the change date. Finally, with respect to the geometries before the changes, in *Segregations* and *Dissolutions* they can be directly obtained through the union of those in the involved domains. In the rest of the cases, the changes in the geometries have to be provided as part of the change information to be able to generate the older geometries.

The applied changes are removed from the set and the matching process is applied again to the remaining ones until the change set is empty or no more changes can be applied. This is needed due to dependencies between changes. Some changes cannot be applied until other changes are integrated. For example, if a jurisdictional domain change references to other jurisdictional domains created/modified at the same date.

The names left unmatched are considered typographical errors, misspellings or variants of official names. In these cases, the *Levenshtein* distance between the names is used to identify the closest equivalence (*getLevJDInChange*). Since the Levenshtein distance does not guarantee a correct selection, to maintain the match quality, we provide a console-based interface which allows reviewing the Levenshtein matches and correct them manually (*reviewMatches*).

Using this algorithm, the processing of the change described in Figure 5 would be performed as follows. First, the domains that persist after the change (*Almodovar del Pinar*, *Chumillas* ...) are located in the knowledge base comparing the names and partof relations (*Cuenca*) with the ones in the knowledge base. After this, since the change is a dissolution, the initial-instant of the involved domains is set to 1983-10-6. A new jurisdictional domain corresponding to *Almodovar de Monterrey* is added to the knowledge base with final instant to 1983-10-6, coupled with a geometry that is the union of the geometries of the other jurisdictional domains.

3. Creation of the Jurisdictional Domain Knowledge Base of Spain

The described creation process has been used in the Virtual Spain project to generate a knowledge base containing the jurisdictional domains of Spain from 1830 to 2011. Virtual Spain is an R&D project supported by the Spanish Government through the Centre for the Development of Industrial Technology whose objective is to define architectures, protocols and standards for an envisioned 3D Internet focusing specially in 3D visualization, virtual worlds, user interactions, and the introduction of semantic capabilities.[1] Florczyk *et al.* (2012) describe the need of constructing a catalogue of orthoimages and associated applications in this project to facilitate the access to geospatial information in Spain. This catalog was extended with historical information to facilitate the discovery of resources. In particular, a historical jurisdictional knowledge base was used to describe properly the places that may be used in the spatial description of the extent of an orthoimage.[2]. This presentation shows the application in which the Spanish jurisdictional knowledge base has been used.

This section details the application of the proposed process to create a historical jurisdictional knowledge base for Spain. Section 3.1 describes the types of Spanish jurisdictional domains that define the knowledge base schema. Section 3.2 describes the sources, their provenance, and structure. Finally, Section 3.3 shows the results of the process, the issues found, and the content of the knowledge base generated.

3.1. *Extension of the Jurisdictional Domain Ontology*

The JDO ontology has been extended with the types of jurisdictional domains that were relevant in the Virtual Spain project. They are the following ones: *comunidad-autonoma* (autonomous community), *comunidad-foral* (statutory community), *ciudad-autonoma* (autonomous city), *provincia* (province) and *municipio* (municipality).[3] A municipality is *part-of* a province, which is *part-of* an autonomous community or a statutory community. Autonomous cities are a specific type of jurisdictional domain used for the Spanish possessions in Africa (*Ceuta* and *Melilla*). The province and the municipality types existed along the whole analyzed period, the rest of them were defined in 1978.

3.2. *Selection of jurisdictional domain instance sources*

The sources that provide the description of the jurisdictional domains and their changes have been selected according to their content quality, structure and reliability of the provider organization. They are provided by official institutions and combining them it is possible to generate the input required for the process to work. Table 1 describes

[1] http://www.xn--espaavirtual-dhb.org/

[2] `http://www.youtube.com/watch?v=gcc4bSJBe9I&feature=youtu.be`

[3] https://raw.github.com/stjdo/stjdo/master/SpanishJurisdictionalDomainOntology.owl

the source providers, the content (CCAA refers to *autonomous community*, *statutory community*, and *autonomous city*), the data format, the period covered by the data, the number of records contained, and the kind of identifier used to tag the content.

Table 1. Collection of files containing parts of the jurisdictional domain information

	Organization	Content	Format	Period	Records	Identifier
Files containing administrative information of current jurisdictional domains						
1	INE	CCAA	Text	2011	19	INE CCAA
2	INE	Provinces	Text	2011	52	INE Prov
3	INE	Municipalities	Excel	2011	8,117	INE Prov+Mun
Files containing the temporal evolution of jurisdictional domains						
4	—	CCAA	Excel	1830-1990	19	SALB CCAA
5	SALB	CCAA	Excel	1990-2009	19	SALB CCAA
6	—	CCAA	Excel	2009-2011	19	SALB CCAA
7	—	Provinces	Excel	1830-2000	52	SALB Prov
8	SALB	Provinces	Excel	2000-2009	52	SALB Prov
9	—	Provinces	Excel	2009-2011	52	SALB Prov
10	REL	Municipalities	Access	1830-2008	19,227	Label
11	INE	Municipalities	Text	2008-2011	42	INE Prov+Mun
Files containing spatial information of jurisdictional domains						
12	BCN	CCAA	Shape	2011	19	INE CCAA
13	BCN	Provinces	Shape	2011	52	INE Prov
14	BCN	Municipalities	Shape	2011	8,188	INE Prov+Mun
Files containing relations between elements in previous files						
15	INE	Provices+CCAA	Text	2011	52	Label
16	SALB	CCAA	Excel	2009	19	SALB+INE CCAA
17	SALB	Provinces	Excel	2009	52	SALB+INE Prov

The first section shows the documents containing administrative information of current jurisdictional domains (year 2011). They have been obtained from the Spanish National Institute of Statistics (Instituto Nacional de Estadistica - INE (2013)). They contain the jurisdictional domain names and the official codes that identify them. The *part-of* relations between provinces and CCAA are provided as pairs of corresponding labels. The relations between municipalities and provinces are inferred from the codes used to identify the municipalities.

The second section provides the files with the changes of jurisdictional domains from 1830 to 2011. The changes in CCAA (from 1990 to 2009) and in provinces (from 2000 to 2009) have been taken from SALB project (Ebener *et al.* 2004). The correspondence between jurisdictional domains in each set, the detection of modifications, and their type of change must be inferred from the structure of the files. The CCAA and province changes in the rest of the 1830-2011 period have been collected through the review of administrative and history related documents. Jurisdictional domains in SALB files can be matched with those in the current model thanks to the correspondences provided. Municipality changes have been obtained from the Spanish Record of Local Entities (Registro de Entidades Locales - REL) and INE. REL provides a database with the jurisdictional domain changes between 1830 and 2008, and INE the list of changes between 2008 and 2011. The sources indicate when each jurisdictional domain change has happened, the type of change, the change date, and the name of involved jurisdictional domains.

The third section includes the documents with the spatial features of the jurisdictional domains. In this case, it was not possible to find a set of documents containing the required spatial information. There are projects such as the China Historical GIS (Berman 2007), and the Atlas of Historical Country Boundaries (Long 1992), which provide very complete spatial historical information about the countries they are focused on (China and United Stated). However, there is nothing similar for Spain. The best options identified are the FAO Global Administrative Unit Layers, which provides the boundaries of the main world jurisdictional domains between 1990 and 2009;[1] and the official Cartographic Reference Base of Spain (Base Cartográfica Numerica - BCN),[2] which contains layers of jurisdictional domain geometries since 1995. To show an example of how the spatial information would be integrated into the ontology, we have added the geometries of current jurisdictional domains using the last BCN version (1:200,000 scale) as source.

The last section contains auxiliary files that relate the identifies of the elements in the previous files, such as the identifies in the SALB files, and their equivalents in the INE files.

3.3. *Population of the knowledge base with current Spanish jurisdictional domains*

The knowledge base has been populated with current domains using the administrative and spatial information provided by files 1-3 and 12-15 in table 1. Altogether, they provide the official name, a unique identification code (created by the INE), the jurisdictional domain geometries, and the *part-of* relations between the jurisdictional domains. Figure 7 shows the transformation of a subset of the source files into the input model. It also remarks the overlapping fields used by the merging process to combine the fragments into complete jurisdictional domain descriptions.

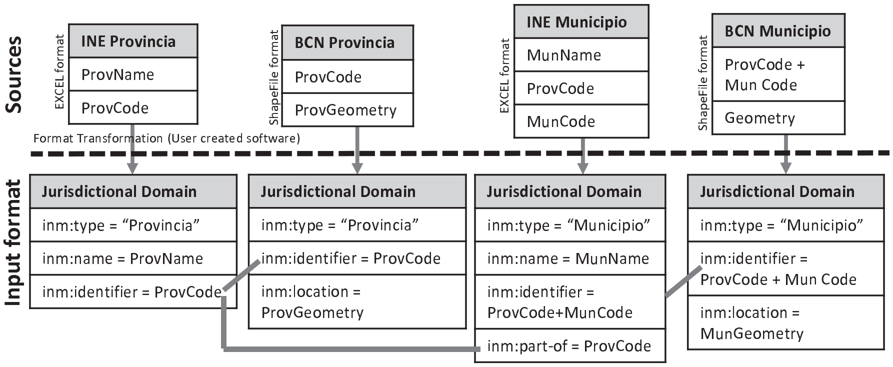


Figure 7. Transformation of current jurisdictional domain source files into the input model

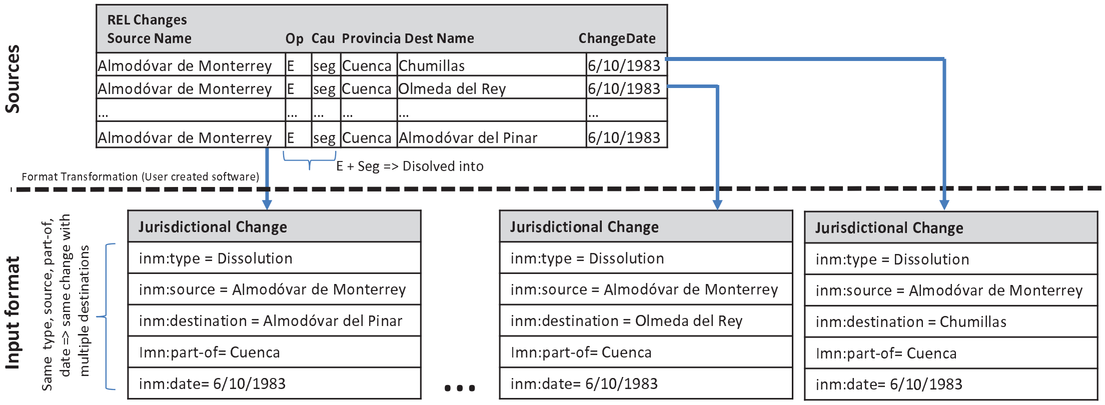
The only issue identified in the data during the integration process was the existence of geometries (in the BCN Municipio file) that could not be matched with any of the generated jurisdictional domains. A revision showed that they correspond to areas that are not part of any municipality (e.g., communal forests in Navarra autonomous community). These entities are not jurisdictional domains and have no correspondence with the extracted administrative names. Therefore, these geometries have been discarded.

[1] `http://www.fao.org/geonetwork/srv/en/` (Administrative and Political Boundaries section)
[2] `http://www.fomento.es/MFOM/LANG_CASTELLANO/DIRECCIONES_GENERALES/INSTITUTO_GEOGRAFICO/CARTOGRAFIA/BBDD/BCN200/default.htm?lang=en`

3.4. *Population of the Spanish jurisdictional ontology with historical jurisdictional domains and their evolution*

The knowledge base has been completed with the changes extracted from the files 4-11 and 16-17 of table 1 (jurisdictional domain changes from 1830 to 2011). SALB files describe CCAA and province changes. Each column shows a snapshot of the jurisdictional domains provided for the period between two changes. REL sources describe the municipality changes. Each change entry includes: operation (*creation* -C-, *update* -M-, and *extinction* -E-), cause (*denomination* -den/and-, *fusion* -fus-, *segregation* -seg-. . .), source and target name of the affected municipalities, province of source municipality (if the destination is in a different province, the municipality name includes its province between brackets), the date of change, and the source of the data that offers this information. In this structure, a change is described by several records (which describe the change from the point of view of each of the involved municipalities).

The extractor created for SALB files obtains the list of changes by identifying the differences between consecutive columns. Each column indicates the state of the model at a given date, so the differences are the changes. With respect to the REL processor, Figure 8 shows an example of conversion in detail. It describes how the dissolution of *Almodovar de Monterrey* municipality in 1983 is transformed from the REL format to the desired input model. The resulting fragments can be integrated by the process thanks to their common fields. (i.e., correspondence in type, source, part-of, and date).



Sources
Format Transformation (User created software)
E + Seg => Disolved into
Input format
Same type, source, part-of, date => same change with multiple destinations
...

REL Changes

Source Name	Op	Cau	Provincia	Dest Name	ChangeDate
Almodóvar de Monterrey	E	seg	Cuenca	Chumillas	6/10/1983
Almodóvar de Monterrey	E	seg	Cuenca	Olmeda del Rey	6/10/1983
…	…	…	…	…	…
Almodóvar de Monterrey	E	seg	Cuenca	Almodóvar del Pinar	6/10/1983

Jurisdictional Change

inm:type = Dissolution
inm:source = Almodóvar de Monterrey
inm:destination = Almodóvar del Pinar
Imn:part-of= Cuenca
inm:date= 6/10/1983

Jurisdictional Change

inm:type = Dissolution
inm:source = Almodóvar de Monterrey
inm:destination = Olmeda del Rey
Imn:part-of= Cuenca
inm:date= 6/10/1983

Jurisdictional Change

inm:type = Dissolution
inm:source = Almodóvar de Monterrey
inm:destination = Chumillas
Imn:part-of= Cuenca
inm:date= 6/10/1983

Figure 8. Example of dissolution conversion into the input model

In the transformation process, we have identified three types of changes not supported in the JDO: *barter*, *segregation with dissolution*, and *partial incorporation with dissolution*.[1] They have been modeled as a composition of other types of changes. A *barter* is a symmetric partial incorporation between territories in which all the jurisdictional domains involved in the change lose part of its surface and gain a part of another one. It has been modeled as sets of partial incorporations. A *segregation with dissolution* is a special case of dissolution in which part of the surface of the dissolved jurisdictional domain is left unassigned (e.g., by the construction of a dam). It has been modeled as a dissolution. Finally, a *partial incorporation with dissolution* is also a special case of incorporation in which the partial incorporations make the source jurisdictional domain to disappear. It

[1] The sources also describe the changes of capital in the municipalities (60 changes) and changes in the capital names (214 changes). However, since they refer to jurisdictional domains of types not included in the model, they have been ignored.

has been modeled as a set of partial incorporations followed by a dissolution that does not generate a new jurisdictional domain.

The transformation of the source content to the input model has shown 35 inconsistencies out of 27,415 records. These inconsistences are caused by misspellings in the record properties (type of change, date, name of the involved municipalities), or changes that do not follow the representation structure (due to missing records or empty fields). Additionally, the integration process has identified more than a hundred of misspellings (e.g., “Arrazua” vs. “Arrazúa”), coordination (e.g., “La Muela” vs. “Muela, La”) and unidentified variants (e.g., “Ajalvir (Daganzo de Abajo)” vs. “Daganzo de Abajo”) issues that have been possible to correct. However, it also showed around nine hundred structural issues in changes around year 1857, which would require deep corrections in the sources to solve them. These issues are caused by changes that generate municipalities not referenced anymore (in other changes or in the current model). A review of a few of them led us to identify them as current minor entities inside other municipalities. However, the sources do not indicate when these municipalities were transformed into minor entities. These errors could be manually corrected by adding the missing information. Nonetheless, we have discarded them because the difficulty of locating their correct evolutions.

3.5. *Analysis of the generated knowledge base*

The generation of the knowledge base took 1 hour and 43 minutes in a single core computer (3Ghz) using around 1GB of RAM memory. It is distributed in three different files whose combination provides the desired jurisdictional evolution model (a total size of 88 MB).[1] The first file contains the current jurisdictional domains (official names and hierarchical relations),[2] the second provides the geometries,[3] and the third has the jurisdictional domain changes.[4] The collection is divided in this way to facilitate its review and use. Current jurisdictional domains are more frequently used than historical ones, so its separation facilitates its use. The same happens with the geometries. Excluding the temporal information (creation, dissolution and change dates), which is managed using reification, the knowledge base only uses the basic elements of OWL language to model jurisdictional information (AL expressiveness).

The whole collection contains around 320,000 statements that describe 14,613 jurisdictional domains: 8,188 of them correspond to current ones (16 autonomous communities, 1 statutory community -Navarra-, 2 autonomous cities -Ceuta and Melilla-, 52 provinces, and 8,117 municipalities), and the rest of them are the extinct ones. As indicated in the ontology used as schema, each jurisdictional domain contains the period in which these jurisdictional domains have existed, their official names along the history tagged with the time-span each label has been official, their geometries if they are available in the sources, and the evolution relations with other domains along with the date these changes have happened. Table 2 shows the number of changes of each type contained in the collection.

The analysis of the generated instances has allowed us to determine how much the jurisdictional model of Spain has changed each year in the time span covered by the generated knowledge base. Figure 9 shows the number of jurisdictional changes per year. It can be observed that the model is not stable. More than two thirds of the Spanish

[1] The generated instances are available at `https://github.com/stjdo/stjdo`

[2] File *SpanishJuridictionalDomainOntology_Admin.owl*

[3] File *SpanishJuridictionalDomainOntology_Spatial.owl*

[4] File *SpanishJuridictionalDomainOntology_Temporal.owl*

Table 2. Number of changes per type

	Name	Incorporation	Partial incorporation	Segregation	Fusion	Dissolution
Number of changes	4,705	3,643	556	472	457	22

jurisdictional model has changed in the covered period and there have been changes almost every year. This corroborates the need of the generated model. If a system dealing with historical data of Spain wants to be precise in the location references used to describe the resources, it can not use a single snapshot of the jurisdictional model due to the frequent changes. To allow the identification of the domains at a given date and their evolution, it needs to model the jurisdictional domains as a continuum.

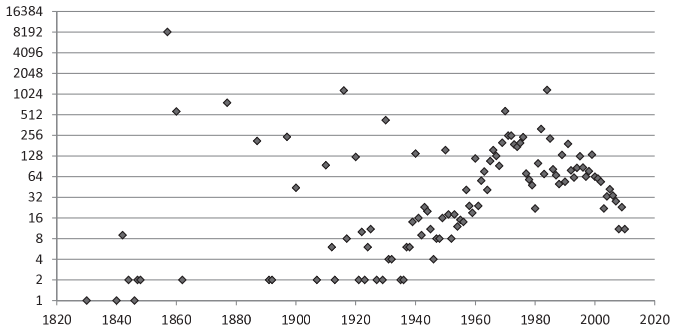


Figure 9. Number of changes in the Spanish jurisdictional domains since 1830

Finally, it is interesting to remark that the process followed to generate the knowledge base has served to create a consistent model. Every jurisdictional domain generated has a time span indicating when it has existed. They all have an (or a set of) official name covering all their existence period. And there are not gaps in the temporal relations between the jurisdictional domains. For example, in all the cases a jurisdictional domain becomes extinct, there is a relation that indicates the reason of the extinction and the jurisdictional domains that absorb its extension.

4. Related work

Focusing on the transformation of geo-spatial information into a semantic model, Goodwin *et al.* (2008) describes the construction process of an administrative gazetteer of Great Britain in RDF. As data source they use BoundaryLine, a collection of geospatial data in raster and vector format. The generation is performed with the help of GIS software that provides the necessary topological relations. Another related approach is the proposal of Auer *et al.* (2009) to publish geospatial information extracted from relational databases as Linked Data. The process is based on mapping HTTP-URI requests onto relational database queries and converting the results into RDF statements. Similarly, Vilches-Blazquez and Corcho (2009) propose a transformation of geographical databases from various Spanish institutions into linked data. They provide a characterization of the types found and describe a heuristic approach to deal with duplicity or co-reference problems.

Berman and Ahlfeldt (2012) is an example of population work process that deals with historical information. It proposes different matching algorithms to link historical place names to gazetteer web services with current jurisdictional domain information. The work focuses on identifying correspondences using labels and coordinate based matching. The work of Sinko (2004) is also relevant. It describes the digitalization of the Atlas of Historical County Boundaries of the USA, and it provides the USA country boundaries since 1864 as yearly snapshots. Although changes are not explicitly depicted, they can be obtained comparing the different snapshots.

With respect to the population of spatio-temporal jurisdictional knowledge bases, Aucott *et al.* (2008) proposes a process to populate a relational model with the jurisdictional information from Sweden, Estonia and United Kingdom. The population process performs the conversion of the source databases using SQL scripts. Kauppinen *et al.* (2008) describes another process to define and populate with instances a spatio-temporal ontology. In this case, the authors focus on Finland jurisdictional domains and its evolution between 1865 and 2007. They populate the defined ontology using a conversion software that processes a set of Excel files with jurisdictional information. Another approach is described in Gantner (2011) and Gantner *et al.* (2012). It shows the construction process of a spatio-temporal ontology of Switzerland covering the period of 1960-2010. The population process converts a relational database into RDF.

5. Discussion

This section compares our proposal with Aucott *et al.* (2008) -QVIZ-, Kauppinen *et al.* (2008) -FSTO-, and Gantner (2011) -SONADUS- works from the related work section (the closest works to this paper proposal).

In QVIZ, the population of the Estonian, Sweden and United Kingdom models is performed in a different way for each case. United Kingdom model is populated merging different databases and manually correcting the inconsistencies. The Swedish model is populated using a single database, and an equivalence map between the source and destination structure. The Estonian model is populated integrating a database (manors), a textual list (parishes), and a collection of shapefiles (geometries). The integration is manually done using database and GIS tools. With respect to FSTO, the population is performed with an automated process that transforms manually generated Excel files, containing the description of the jurisdictional domains, into RDF. Finally, SONADUS is populated from a GIS database and a collection of Excel and Shape files. They import the sources into a database and merge them through SQL queries.

It can be observed that each population approach is closely tied to the source structure. Only the FSTO process defines a common input format for the data to integrate. However, it does not analyze the problematic of generating this input format from other existent sources.

Our approach divides the process to populate a jurisdictional knowledge base into tasks dependent on the source format, and tasks independent from the source format. The dependent tasks are different for each source, but the independent ones can be reused in different population processes. Thanks to this, the work required to generate a new jurisdictional knowledge base is reduced to: (1) the definition of the knowledge base schema (extending JDO); and (2) the transformation of the sources into the format accepted by the reusable integration components. For example, in QVIZ or SONADUS this would avoid the need to define the database operations that perform the source integration. It would require instead the conversion of each source into the input format, but we think this is simpler as it does not need data integration.

With respect to the knowledge base schema, JDO is complete enough to model the countries jurisdictional information and their evolutions. Table 3 shows the correspondence of the main JDO relations with QVIZ, FSTO, and SONADUS models. It can be observed that they are compatible excepting the Establishment/Initial entry changes. However, this is not an issue. This relation is used to describe the first reference to a jurisdictional domain in the data collection, not the real creation date of the jurisdictional domain.

Table 3. Correspondence in change types between the different ontologies

JDO	QVIZ	FSTO	SONADUS
Fusion		Merge	Merger
Incorporation	Abolished to enlarge	Changepartof	Annexation
Partial incorporation	Reduced to enlarge		Land exchange
Segregation	Reduced to create	Changepartof	Secession
Dissolution	Abolished to create	Split	Separation
		Establishment	Initial entry
Name			Change of Name
Part-of	IsPartOf	Changepartof	Change of . . .

With respect to the validation of the generated models, the existent approaches are heterogeneous. In QVIZ, a hierarchical validation is performed when the model is constructed. Inconsistences are logged so they can be manually reviewed. They also use graphical tools to manually review the jurisdictional domain geometries. In FSTO, the sources quality is assumed as they do not describe how the inconsistences are managed. In SONADUS, the errors detection is performed though the revision of maps showing the changes. Our validation approach goes in the line of QVIZ automatic detection, but extended to other types of possible errors. On the one hand, the merge of the source fragments detects elements that are not merged due to the lack of equivalences. On the other hand, the change integration detects those changes that do not fit in the model (e.g., there are not jurisdictional domains with the names indicated in the change). When these inconsistences are identified, the user is informed so that he can take correcting measures.

6. Conclusions and future work

This paper has described a process to create and populate a knowledge base that represents the evolution of the jurisdictional domains of a country from a date to present. Specifically, it has described how to extend the ontology used as schema of the knowledge base with the types of jurisdictional domains of the country, and how to extract from available resources the information required to populate the knowledge base.

The process separates the extraction of jurisdictional information from the sources, and its integration into the knowledge base. This allows the reuse of the integration components and reduces the amount of software required to implement to perform a transformation. This separation has required the definition of a common input model that stores the source content as a collection of changes and a snapshot of the jurisdictional domains after the last change. The information about a change or a jurisdictional domain from the snapshot can be fragmented in several input resources to simplify the transformation of the sources into the input model. The generation process combines the fragments and uses them to generate the desired jurisdictional knowledge base.

The process has been applied to create a knowledge base containing the evolution of the Spanish jurisdictional domains from 1830 to 2011, which was successfully integrated in the Virtual Spain project (see Section 3) for the development of an orthoimages catalogue with support to searches based on jurisdictional domains. First, we have made a review of the characteristics of the data used as source. Then, it has been shown how the generation process is applied, the complexity of the transformation and integration of the source information, and the issues related with the detection and correction of inconsistences. Finally, the generated knowledge base has been analyzed and some measures about the evolution of the Spanish jurisdictional domains have been provided.

The application of this process to other source collections should be quite straightforward as long as they fulfil the prerequisites indicated in sections 2.1 and 2.3.1: their content can be transformed into a collection of current jurisdictional domains plus a list of changes; and the overlap between the sources allows the automatic integration by the proposed algorithm.

Some of our future work lines rely on the use of this process to generate new jurisdictional domain models. On the one hand, we want to complete the ontology with the types of jurisdictional domains not currently included, and with the geometries of the historical jurisdictional domains. This extension will allow us to determine how well the generation process deals with data sources containing types of jurisdictional domains different from the currently used. Additionally, we want to work on the integration of jurisdictional domains from different countries. The objective is to analyze how the types of jurisdictional domains of different countries can be matched to support location-based queries in border areas. Therefore, we need to generate jurisdictional models from Spain neighboring countries. This will let us test the process with quite different jurisdictional type structures and source formats. Finally, to facilitate the maintenance of the generated models up to date, we want to allow the addition of new jurisdictional domain changes to an existing knowledge base without having to reconstruct it from scratch. The idea is to define an additional process step able to add the new changes as soon as they are published.

Acknowledgments

This work has been partially supported by Spanish Government (TIN2012-37826-C02-01), the National Geographic Institute (IGN) of Spain and GeoSpatiumLab S.L.

References

Aucott, P., *et al.*, 2008. *Administrative Unit Ontology Report and Schema D3.2.* Technical report, Umea University.

Auer, S., *et al.*, 2009. Triplify: light-weight linked data publication from relational databases. *In*: *18th international conference on World wide web*, April., Madrid, Spain, 621–630.

Berman, M.L., 2007. Modeling Spatio-Temporal Networks with CHGIS. *In*: *2nd International Workshop on Monies, Markets, and Finance in China and East Asia*, October., Bochum, Germany.

Berman, M.L. and Ahlfeldt, J., 2012. Historical Gazetteer Development and Integration. *In*: *Working Digitally with Historical Maps*, February., New York, USA.

Bizer, C. and Cyganiak, R., 2006. D2R Server - Publishing Relational Databases on the

Semantic Web. *In*: *5th International Semantic Web Conference*, November., Athens, Greece.

Carr, A., 2006. Expansion of Prussia 1807-1871. [online] `http://en.wikipedia.org/wiki/File:Ac.prussiamap3.gif` [January, 2014].

Cotelo Lema, J.A., 2006. *External Representation of Spatial and Spatio-Temporal Values.* Technical report, Fernuniversität inHagen.

Ebener, S., Morjani, Z.E., and Guigoz, Y., 2004. The Second Administrative Level Boundaries data set project (SALB). A working platform for improving data sharing. *In*: *4th EnviroInfo Conference*, October., Geneva, Switzerland.

Ellson, J., *et al.*, 2002. Graphviz - Open Source Graph Drawing Tools. *Lecture Notes in Computer Science*, 2265, 483–484.

Florczyk, A., *et al.*, 2012. Identifying Orthoimages in Web Map Services. *Computers and Geosciences*, 47, 130–142.

Gantner, F., 2011. A Spatiotemporal Ontology for the Administrative Units of Switzerland. Thesis (PhD). University of Zurich.

Gantner, F., *et al.*, 2012. The basic formal ontology as a reference framework for modeling the evolution of administrative units. *Transactions in GIS*, 17 (2), 206–226.

Goodwin, J., Dolbear, C., and Hart, G., 2008. Geographical Linked Data: The Administrative Geography of Great Britain on the Semantic Web. *Transactions in GIS*, 12, 19–30.

Gregory, M., *et al.*, 2011. Domain Independent Knowledge Base Population From Structured and Unstructured Data Sources. *In*: *Twenty-Fourth International Florida Artificial Intelligence Research Society Conference*, USA., Palm Beach, 251–256.

Guarino, N., 1998. Formal Ontology and Information Systems. *In*: *First International Conference on Formal Ontologies in Information Systems*, June., Trento, Italy, 3–15.

Hart, G. and Dolbear, C., 2013. Publishing linked data - Linked data generation. *In*: *Linked Data: A Geographic Perspective.*, 115–122 CRC Press.

INE, 2013. List of Spanish provinces and municipalites. Instituto Nacional de Estadísticas. [online] `http://www.ine.es/daco/daco42/codmun/codmun06/06codmunmapa.htm` [April, 2013].

ISO, 2003. *Geographic information – Spatial referencing by geographic identifiers.* Technical report, International Organization for Standardization, ISO/TC 211.

Kauppinen, T. and Hyvönen, E., 2007. Integrated Series in Information Systems, Modeling and Reasoning about Changes in Ontology Time Series. *In*: *Ontologies.*, 319–338 Springer.

Kauppinen, T., Väätänen, J., and Hyvönen, J.E., 2008. Creating and using geospatial ontology time series in a semantic cultural heritage portal. *In*: *5th European semantic web conference on The semantic web: research and applications*, June., Tenerife, Spain, 110–123.

Lacasta, J., *et al.*, 2013. Design and evaluation of a semantic enrichment process for bibliographic databases. *Data & Knowledge Engineering*, 88, 94–107.

Lacasta, J., Nogueras-Iso, J., and Zarazaga-Soria, F.J., 2010. Ontology basic concepts. *In*: *Terminological Ontologies. Design, Management and practical applications.*, 1–24 Springer.

Long, J.H., 1992. A Case Study In Utilizing Computer Technology: The Atlas of Historical Country Boundaries. *Perspectives: American Historical Association Newsletter*, 30, 16–17.

López-Pellicer, F.J., *et al.*, 2011. An Ontology for the representation of Spatio-Temporal Jurisdictional Domains in Information Retrieval Systems. *International Journal of*

Geographical Information Science, 26 (3), 579–597.

Magnini, B., *et al.*, 2006. Ontology Population from Textual Mentions: Task Definition and Benchmark. *In*: *2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, July., Sydney, Australia, 26–32.

Maynardand, D., Li, Y., and Peters, W., 2008. Vol. 167 of *Frontiers in Artificial Intelligence and Applications*, NLP Techniques for Term Extraction and Ontology Population. *In*: *Ontology Learning and Population: Bridging the Gap between Text and Knowledge.*, 107–127 IOS Press.

Murray, C., 2006. *Oracle Spatial. Users Guide and Reference. 10g Release 2 (10.2).* Technical report, Oracle Corp., Redwood Shores, CA.

Özçep, O.L., Grütter, R., and Möller, R., 2012. Dynamics of a Nearness Relation - First Results. *In*: *STeDy 2012: International Workshop on Spatio-Temporal Dynamics*, August., Montpellier, France, 1–7.

Pridal, P. and Zabicka, P., 2008. Tiles as an approach to on-line publishing of scanned old maps, vedute and other historical documents. *e-Perimetron*, 3 (1), 10–21.

Rodriguez, J.B. and Gómez-Pérez, A., 2006. Upgrading relational legacy data to the semantic web. *In*: *15th international conference on World Wide Web*, May., Edinburgh, UK, 1069–1070.

Sinko, P.T., 2004. The Atlas Goes Digital. *Origins*, 1, 4–5.

Stojanovic, L., Stojanovic, N., and Volz, R., 2002. Migrating data-intensive Web Sites into the Semantic Web. *In*: *ACM Symposium on Applied Computing*, March., Madrid, Spain, 1100–1107.

Tappolet, J. and Bernstein, A., 2009. Applied Temporal RDF: Efficient Temporal Querying of RDF Data with SPARQL. *In*: *6th European Semantic Web Conference*, Vol. 5554 of *Lecture Notes in Computer Science*, May., Heraklion, Greece, 308–322.

Vilches-Blazquez, L.M. and Corcho, O., 2009. A heuristic approach to generate good-quality linked data about hydrography. *In*: *20th International Workshop on Database and Expert Systems Application*, August., Linz, Austria, 99–103.

Villazón-Terrazas, B., *et al.*, 2011. Methodological Guidelines for Publishing Government Linked Data. *Linking Government Data*, 27–49.

W3C, 2013. *SPARQL 1.1 Overview.* Technical report, World Wide Web Consortium.

Weiten, M., 2009. OntoSTUDIO as an ontology engineering environment. *In*: *Semantic knowledge management.*, 51–60 Springer.

Zimmermannc, A., *et al.*, 2012. A General Framework for Representing, Reasoning and Querying with Annotated Semantic Web Data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 11, 72–95.