Mining for Connections in the Records: Knowledge Graphs in Beyond 2022

Christophe Debruyne

The Beyond 2022 project aims to create a virtual archive by digitally reconstructing and digitising historical records lost in an explosion and destroyed by fire in the Public Records Office of Ireland in 1922. The project is developing a knowledge graph to facilitate information retrieval and discovery over the reconstructed items. In this article, we cover some of the design considerations of our knowledge graph and demonstrate how the knowledge graph can be used to explore the information "siloed" in traditional documents.

The examples in this article are based on sample data drawn from the index to Irish Exchequer Payments 1270-1326, ed. Philomena Connolly.

What is a Knowledge Graph?

A knowledge graph (KG) is said to be a set of interconnected typed entities and their attributes and relationships (Pan et al., 2016). Take, for instance, the example in Figure 1. In that figure, the attributes and relationships are captured as directed edges, and entities are represented as ellipses. Entities represent things that cannot be "printed on a screen". Attributes, on the other hand, are things that can be meaningfully "printed on a screen" and such as names, labels, and titles. If we were to look closer at the graph, we notice that the graph describes an entity, Ralph Bristol, which is an instance of the type Person. We also have an entity Bishop of Kildare, which is of an instance of the type Group and represents the groups that are of the type Rank. The graph furthermore captures that Ralph Bristol was a member of that group and that the label of the rank Bishop of Kildare is "Bishop of Kildare". The types and relations are provided by

vocabularies. Before we can elaborate on the exact nature of vocabularies, we will first describe our KG's data model.

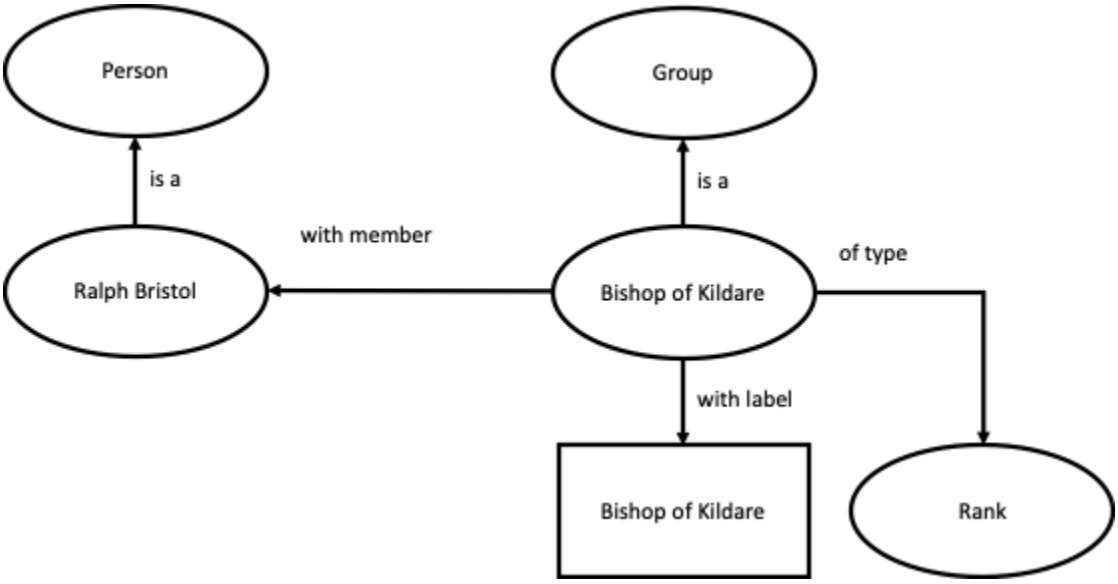

Figure 1: an example of a knowledge graph.

Knowledge Graphs in Beyond 2022

The Beyond 2022 KG is stored with the Resource Description Framework (RDF), which is a W3C Recommendation (i.e., a standard) for representing and linking structured information on the Web.[1] In RDF, graphs are constructed with so-called triples that are of the form:

<subject, predicate, object>

The *subject* is always a *resource* (i.e., a thing), which may be identified by a URI. For instance, the entity Bishop of Kildare is identified by the URI http://kb.beyond2022.ie/rank/bishop%20of%20Kildare. The *predicate* used to represent the relationship is always a resource identified by a URI. For instance,

[1] https://www.w3.org/TR/rdf11-primer/, last accessed April 24, 2020

http://www.w3.org/2000/01/rdf-schema#label provides us a predicate to relate things with their label. The *object* of an RDF triple is either a resource (identified by a URI or not) or a literal, e.g., "Bishop of Kildare". The triple used as an example, thus, looks as follows:

<http://kb.beyond2022.ie/rank/bishop%20of%20Kildare,
http://www.w3.org/2000/01/rdf-schema#label,
"Bishop of Kildare">

Since RDF allows us to identify resources with a URI, we can relate our entities with resources described in other RDF datasets, allowing us to integrate information across the Web. Unlike most knowledge graph languages, RDF provides us with a *distributed* graph data model. The reader might have noticed that the predicate for label seemed to have been declared in another domain. Indeed, we have reused an existing resource for relating things to their labels, which brings us to *vocabularies*.

In a KG, the types and relations are usually provided by *vocabularies*.[2] Vocabularies provide a formal, explicit specification of a shared conceptualisation (Gruber, 1995; Studer et al., 1998) and are key in rendering those knowledge graphs meaningful–when one uses recognized and established vocabularies, others can interpret the graphs that one provides. Vocabularies are declared in namespaces, which are identified by a URI.

> *In the remainder of the article, we will use namespace prefixes. Namespace prefixes allow us to relate the URI of a vocabulary to a name and access the named resources of that vocabulary via a shorthand. The predicate label, for instance, is declared in the namespace http://www.w3.org/2000/01/rdf-schema# for which the namespace prefix "rdfs" is typically adopted. So*

[2] Vocabularies are sometimes called ontologies. The latter usually refers to ontologies that contain few axioms–vocabularies are meant to support interoperability whereas ontologies are typically developed for specific reasoning tasks. In other words, vocabularies are "light-weight" ontologies.

rather than "writing" http://www.w3.org/2000/01/rdf-schema#label, we can "write" rdfs:label.

The vocabularies we use for the Beyond 2022 project include:

- *CIDOC-CRM* (Boerz et al., 2008), prefixed with "cidoc". CIDOC-CRM provides a broad set of types and relations for representing cultural sector data. While broad, CIDOC-CRM does not provide us with all the concepts we need. CIDOC-CRM provides us a way to qualify entities with their notion of a type via a predicate cidoc:P2_has_type.
- The types we introduced for our KG were declared in our *Beyond 2022 ontology*, which is prefixed with b2022. Concepts introduced in our ontologies include Rank, Office, and Floruit. Those were all concepts for which abstract types existed (e.g., Group and Period), but needed to be qualified with our terminology as to make those meaningful and distinguishable from others.
- Finally, for the sake of this article, the final two vocabularies are *RDF* and *RDFS*[3]. The RDF data model provides us a predicate to relate things with their type, namely rdf:type. The latter, which is a schema language built on top of RDF, provides us with predicates that are commonly used on the Semantic Web (such as rdfs:label).

Now that we have introduced the vocabularies used in the KG, we can "implement" our abstract KG of Figure 1. In Figure 2, we have replaced the edges' labels with predicates. The types and entities have been replaced with resources from CIDOC-CRM and our Beyond 2022 ontology. We also replaced the things denoting Ralph Bristol and Bishop of Kildare with their corresponding URIs in our knowledge graph. Different colors were used to emphasize in which vocabulary an entity or relation is declared.

[3] https://www.w3.org/TR/rdf-schema/, last accessed April 24, 2020

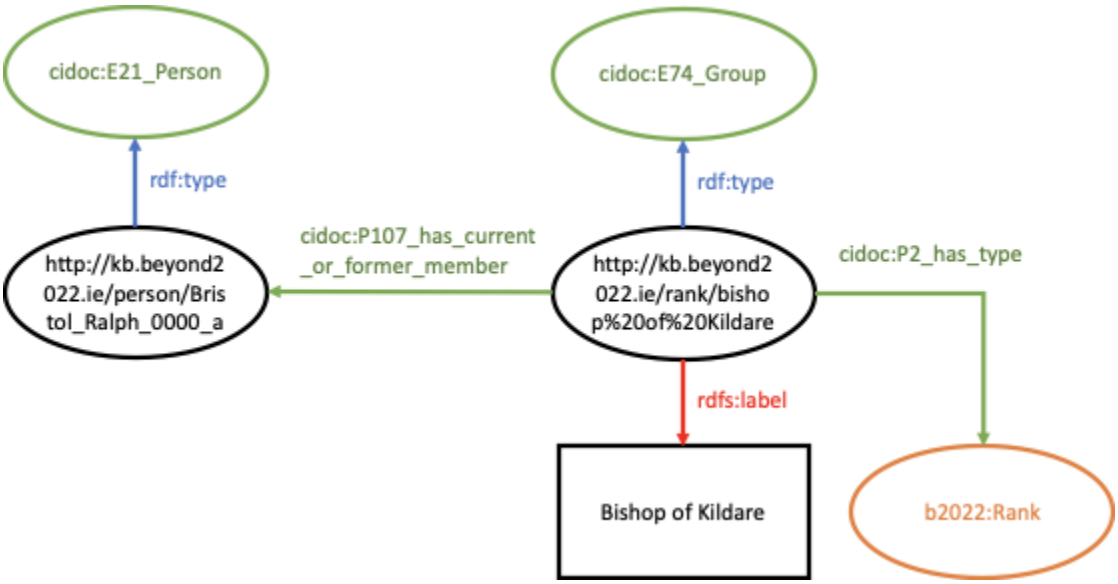

Figure 2: The "implementation" of the abstract KG depicted in Figure 1.

> *The uptake of RDF by memory institutions is known to be problematic, as evidenced by surveys conducted in (McKenna et al., 2018). While there are quite a few noteworthy projects, issues with Semantic Technologies include tooling and the background knowledge required. While not the topic of this article, it suffices to say that subject matter experts in this project capture data in spreadsheets, which are then transformed into more elaborate knowledge graphs.*

Now that we have covered what knowledge graphs are and how Beyond 2022 adopted standardized Semantic Web technologies to represent and store these knowledge graphs, we can explore how one can engage with these.

Engaging with Knowledge Graphs

The advantages of using standards include the existence of third party software, which we can adopt to explore the knowledge graphs–they merely need to "understand" RDF or any of the standardized vocabularies we have adopted. Various tools exist, from faceted browsing and query interfaces to all sorts of visualizations of the knowledge

graph. In this article, we will demonstrate Ontodia (Mouromtsev et al., 2015), which combines a limited form of faceted search with a compelling visual exploration. Ontodia allows one to explore a knowledge graph by looking for and placing entities on a diagram. The diagrams can be expanded by inspecting the relations of entities displayed on the diagram.

In Figure 3, we simply placed an entity representing a person on a diagram and explore some of its relations. On the left, one can look for instances by type (called "classes" in RDF). Entities already appearing on the diagram become disabled (as is the case with the entity cidoc:E21_Person). Also, notice that the person is listed in an authority document identified by a URI from outside the Beyond 2022 system. One can explore and follow that URI to, hopefully, retrieve more information about that person. In this specific case, the user will be directed to an entry in the Oxford Dictionary of National Biography.

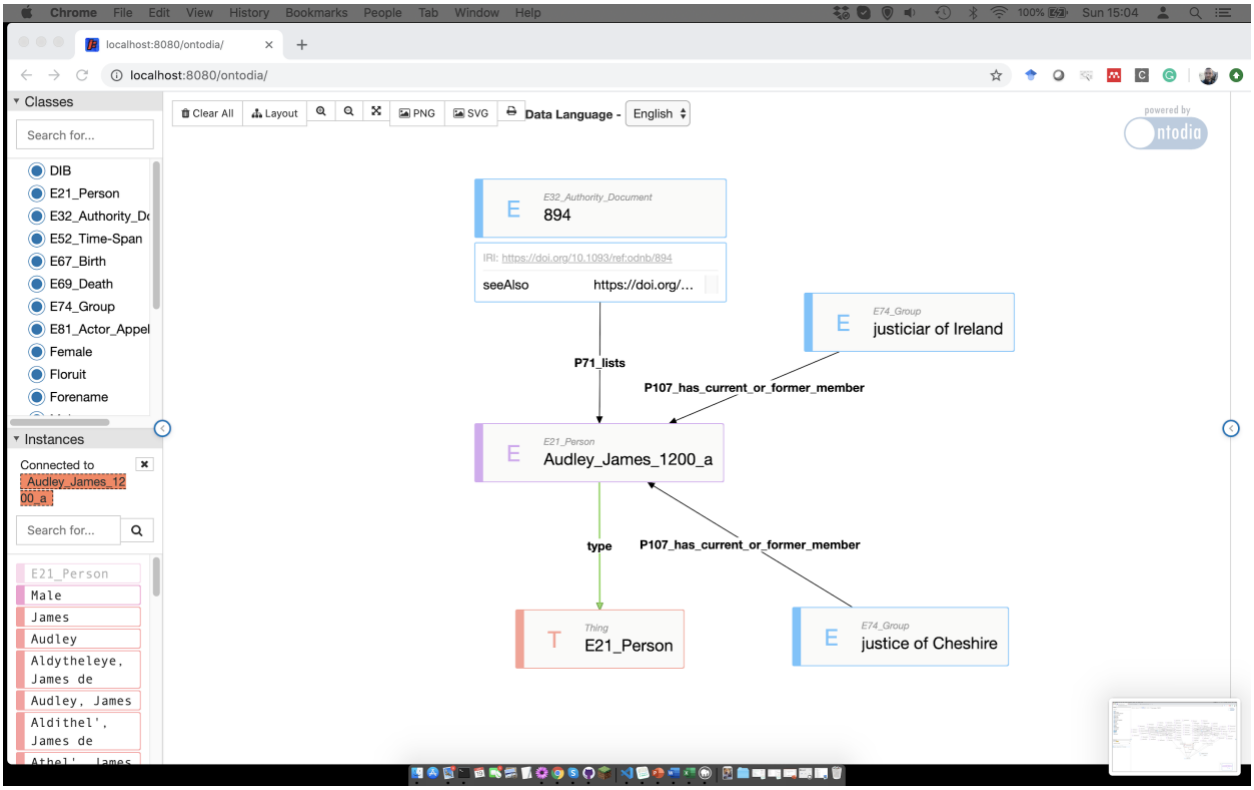

Figure 3: Using Ontodia to explore the Beyond 2022 Knowledge Graph.

The capabilities of the knowledge graph become more apparent when one wants to answer questions that would take a subject matter expert much time searching and reading documents. Assume that we are interested in figuring out who held three specific offices–e.g., justiciar of Ireland, treasurer of Ireland, and chancellor of Ireland–by:

1. selecting the three offices and placing them on the diagram;
2. displaying the people who were affiliated with that office on that diagram; and
3. use a graph layout algorithm to discover the people that held more than one office.

The result of this process is shown in Figure 4. In Figure 4, we see that every pair of offices share some people and that there is, for the data curated from the Irish Exchequer Payments index so far, only one person that held the three offices; Alexander Balscot.

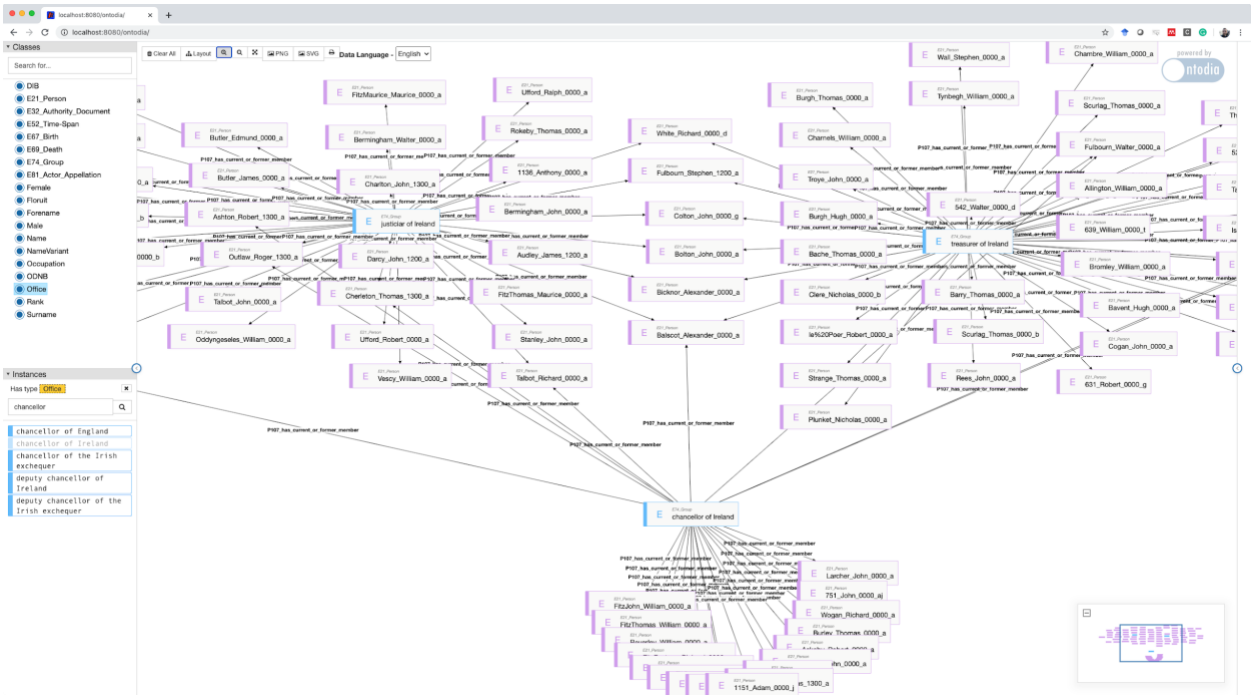

Figure 4: visually exploring the Beyond 2022 knowledge graph with Ontodia.

While not as expressive as querying the knowledge graphs with SPARQL[4], a W3C Recommendation for querying RDF, it is hoped that one can more rapidly discover information contained in documents and historical records. To indicate what would be possible in SPARQL, we first provide the SPARQL query looking for people that shared these three offices, and then a query looking for people that held at least three offices.

In Listing 1, we have the query looking for things that are of the type person and were members of three specific offices (treasurer, chancellor, and justiciar of Ireland). The range of the property "has current or former member" is "Actor", which is a more abstract type than "Person". This is why it is necessary to provide the requirement that the instances bound to the variable ?person have to be of the type "Person" on line 4. Lines 5, 6, and 7 require those instances have to be affiliated with those three offices.

```
1. PREFIX cidoc: <http://erlangen-crm.org/current/>
2. PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3. SELECT DISTINCT ?person WHERE {
4.     ?person rdf:type cidoc:E21_Person .
5.     <http://kb.beyond2022.ie/office/justiciar%20of%20Ireland>
   cidoc:P107_has_current_or_former_member ?person .
6.     <http://kb.beyond2022.ie/office/chancellor%20of%20Ireland>
   cidoc:P107_has_current_or_former_member ?person .
7.     <http://kb.beyond2022.ie/office/treasurer%20of%20Ireland>
   cidoc:P107_has_current_or_former_member ?person .
8. }
```

Listing 1: looking for instances of people that are part of the group that held the offices of treasurer, chancellor, and justiciar of Ireland.

[4] https://www.w3.org/TR/sparql11-query/, last accessed April 24, 2020

While arguably complex, SPARQL is a powerful query language. While Ontodia provides a pleasant way to query the knowledge graph, it uses SPARQL to interrogate the said graph. These tools use the capabilities of that query language to present the user with numbers such as, for each entity, the number of relationships of a particular predicate (see Figure 5).

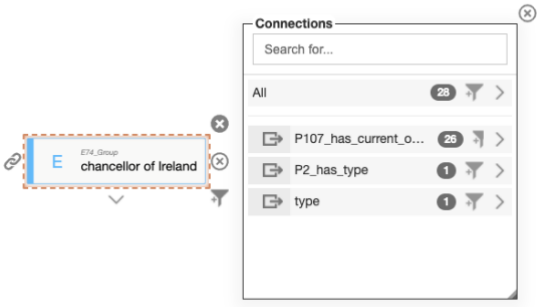

Figure 5: The entity representing the office "chancellor of Ireland" has 26 relationships; one rdf:type relationship, one cidoc:P2_has_type relationship, and 26 relationships with people.

We can avail of these same capabilities to interrogate the knowledge graph. In Listing 2, we formulate a query that looks for people sharing at held three offices, count the number of offices they held, and order them by the number of offices they held.

```
1. PREFIX cidoc: <http://erlangen-crm.org/current/>
2. PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3. SELECT ?person ( COUNT(?office) AS ?count ) WHERE {
a.     ?person rdf:type cidoc:E21_Person .
b.     ?office cidoc:P107_has_current_or_former_member ?person .
4. }
5. GROUP BY ?person
6. HAVING ( COUNT(?office) >= 3 )
```

7. ORDER BY DESC(?count)

Listing 2: Looking for people that held at least three offices and the number of offices they held. The results are ordered by the number of offices in descending order.

Using the knowledge graph to date, we have 142 people that held at least three offices, of which four have held 7. Figure 6 shows some of the results of the query in Listing 2.

Showing 1 to 50 of 142 entries Search: Show 50 entries

	person	count
1	<http://kb.beyond2022.ie/person/Bicknor_Alexander_0000_a>	"7"^^xsd:integer
2	<http://kb.beyond2022.ie/person/Brown_Richard_0000_a>	"7"^^xsd:integer
3	<http://kb.beyond2022.ie/person/Sandford_John_0000_a>	"7"^^xsd:integer
4	<http://kb.beyond2022.ie/person/le%20Poer_Robert_0000_a>	"7"^^xsd:integer
5	<http://kb.beyond2022.ie/person/Bagot_Robert_0000_a>	"6"^^xsd:integer
6	<http://kb.beyond2022.ie/person/Barton_William_0000_a>	"6"^^xsd:integer
7	<http://kb.beyond2022.ie/person/Bridge_Henry_0000_a>	"6"^^xsd:integer
8	<http://kb.beyond2022.ie/person/Exeter_Richard_0000_a>	"6"^^xsd:integer
9	<http://kb.beyond2022.ie/person/1311_John_0000_dc>	"5"^^xsd:integer
10	<http://kb.beyond2022.ie/person/562_Thomas_1300_a>	"5"^^xsd:integer
11	<http://kb.beyond2022.ie/person/631_Robert_0000_g>	"5"^^xsd:integer
12	<http://kb.beyond2022.ie/person/749_John_0000_ai>	"5"^^xsd:integer

Figure 6: Partial account of Listing 2's result set.

Summary

In summary, knowledge graphs allow us to represent the rich and complex information contained in historical records as typed entities and their interrelationships. Knowledge graphs allow us to interrogate and engage with that information to discover and gain insights. Some of the technology behind these knowledge graphs are, arguably, not that accessible to non-experts. The use of standardized knowledge graph technologies, however, allows us to avail of many tools capable of processing these technologies and not necessarily rely on the development of bespoke tools. We have demonstrated this with Ontodia, a generic knowledge graph exploration tool that combines simple faceted search with a rather intuitive interface for visually exploring the graph by creating diagrams.

References

Connolly, P.: Irish Exchequer Payments 1270-1326. Irish Manuscripts Commission (1998).

Goerz, G., Schiemann, B., Oischinger, M.: An implementation of the CIDOC Conceptual Reference Model (4.2. 4) in OWL-DL. Proc. 2008 Annu. Conf. CIDOC Athens, Sept. 15–18, 2008. (2008).

Gruber., T.: Toward principles for the design of ontologies used for knowledge sharing? Int. J. Hum.-Comput. Stud., 43(5-6):907–928, 1995.

McKenna, L., Debruyne, C., O'Sullivan, D.: Understanding the Position of Information Professionals with regards to Linked Data. In: Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries - JCDL '18. pp. 7–16. ACM Press, New York, New York, USA (2018).

Mouromtsev, D., Pavlov, D., Emelyanov, Y., Morozov, A., Razdyakonov, D., Galkin, M.: The Simple Web-based Tool for Visualization and Sharing of Semantic Data and Ontologies. In: Proceedings of the ISWC 2015 Posters & Demonstrations Track co-located with the 14th International Semantic Web Conference (ISWC-2015), Bethlehem, PA, USA, October 11, 2015. CEUR-WS.org (2015).

Pan, J. Z., Vetere, G., Gomez-Perez, J. M., Wu, H. (eds) Exploiting Linked Data and Knowledge Graphs in Large organizations, Springer. (2016).

Studer, R., Benjamins, R., and Fensel, D.: Knowledge engineering: Principles and methods. Data & Knowledge Engineering, 25(1–2):161–198, 1998.