Competency questions for a test first development of an energy systems analysis ontology

Eugenio S. Arellano R.[1], Ulrich Frey[1] and Carsten Hoyer-Klick[1]

[1]German Aerospace Center (DLR), Department of Energy Systems Analysis, Institute of Networked Energy Systems, Stuttgart, Germany

Abstract

The field of energy systems research is highly interdisciplinary and handles large amounts of complex data. This complexity calls for specific data management methods. Formal structures like ontologies are increasingly being used to organize information of this field. However, such methods have to be balanced against application development needs. Thus, it is important to be able to evaluate the usefulness of an ontology. This paper is about translating example research questions into queries. These queries can be used to evaluate the completeness of an ontology. They are an instance of what in ontology development are called competency questions. We focus on their use in test first infrastructures. The output of this study is a collection of nine such queries with a detailed explanation of how they were designed. These results are intended to offer an energy systems researcher strategies to organize their demands in regards knowledge graph design. This may help them perform better data management decisions, which have implications on understandability and performance. We emphasize that knowledge organization in such an interdisciplinary field requires splitting up the effort into smaller research units that can later be unified in a larger knowledge graph.

Keywords

Energy systems modelling, Ontology applications, Data models, Knowldege graphs,

1. Introduction

Energy systems analysis is a highly interdisciplinary field of which software and data is an intrinsic part. Its tools have evolved from linear programming models with an amount of variables easily manageable with spreadsheet software to complex software suites. These use specialized computational methods and solvers to process data such as high resolution (time and space) renewable energy potentials. Energy systems models are a prime example of these tools. They comprise mathematical models that simulate, extrapolate and/or optimize an energy system. These models not only consume large quantities of heterogeneous data but also produce large amounts of (big) data.

The above-mentioned facts justify the research and development of data management techniques specific to the field of energy systems analysis. Chen et al. [1] state that knowledge graphs have moved into focus in the last decade. They are part of the increasing number of tools being used in modern data research. To maximize their research potential and to enable its

The Eighth Joint Ontology Workshops (JOWO'22), August 15-19, 2022, Jönköping University, Sweden

eugenio.arellanoruiz@dlr.de (E. S. A. R.); ulrich.frey@dlr.de (U. Frey); carsten.hoyer-klick@dlr.de (C. Hoyer-Klick)

0000-0003-2508-3976 (E. S. A. R.)

operability in multiple disciplinary contexts, it is necessary to enrich the information it contains with semantics. To do this, ontologies are an important element to be leveraged during the development of these graphs. Choosing an ontology should be done based on the requirements of the applications built on top of the knowledge graph and the type of data to be fed into it.

This paper intends to offer a structured example on how to systematically construct queries to evaluate the usefulness of an ontology in the context of energy systems research. We propose that these are used in an analogous method of what is known in software development as Test Driven Development (TDD). This approach uses what is traditionally known in ontology development as competency questions. What is different from the usual application of the latter is that this method takes an ontology user perspective. This means that practical implementations are considered more important than consistency checks.

1.1. Energy Systems Models

Energy Systems Models are software tools that build representations of an energy system. The characteristics of these representations differ according to the kind of research the model is used for. There are numerous types of energy systems models and categorizing them is hard. Pfenninger et al. [2] propose a four-group categorization. The first are Energy Systems Optimization Models (ESOMs) which consist of optimization methods that provide scenarios of how the system could evolve under fixed conditions. In the second group are simulations which include techniques that forecast changes of the system. The third group are energy market models which focus on market dynamics. The fourth group consists of qualitative or mixed methods. These are more concerned with narratives and heuristics. These categories are not fixed. In reality, models often use a combination of the mentioned methods.

The data used as an input to models is diverse in various dimensions. Spatial resolutions can be so low that whole continents are represented as single nodes and so high that individual components and transmission lines are modelled. The temporal scope may range from seconds to decades. File sizes range from a few kilobytes for parameters to gigabytes for renewable potential time series. Hence, this data represents a very diverse range of entities. On one hand physical things like power plants, transmission lines and storage units have to be represented. On the other hand there are abstract concepts like techno-economic parameters e.g. costs, interest rates, and efficiencies that go into models (DeCarolis et al. [3]). The scale of model output is usually the same as the input. The organization of outputs presents a challenge, especially if they are part of a modelling chain. In this case they are used as inputs for other models. Sometimes, these values need adjustments or post-processing. The complexity of the data used in such models usually precludes a unified view of definitions across different models. This complicates model coupling and is highly non-transparent. According to Henry et al. [4], extensive annotation of such data is not yet widespread, even in openly available energy model data sets.

1.2. Testing the Open Energy Ontology

These concerns led Booshehri et al. [5] to the creation of the Open Energy Ontology (OEO). It has the goal of describing the relevant data and modelling approaches with all their characteristics.

It is in continuous development using agile principles. For example, release cycles comprise gatherings of users and developers that discuss the next steps of feature inclusion. While this approach has pushed the completion of the project forward, the development team has yet to find a way of reporting to which degree the ontology accomplishes its end goal. This shortcoming can be covered by the inclusion of another agile concept: Test First (TF) programming.

TF is another name given to the previously mentioned TDD. As described by Madeyski [6], consists of programmers writing a test before writing an actual piece of production code. Its usage by agile teams is characterized mainly by shifting away from relying on anecdotal evidence as sufficient ground for decision-making to a systematic evaluation of controlled experiments.

Testing of an ontology is done using what is called competency questions. These were conceptualized by Gruninger [7]. In Web Ontology Language (OWL) ontologies like the OEO they are implemented as Description Logic (DL) queries. These are used to test the validity and consistency of an ontology. They are executed using special software called reasoners. The OEO uses a set of fifty competency questions which were used to describe the ontology at the time of its first release. They are part of its existing test infrastructure, and they make sure that the logic behind its original elements remains consistent across versions. It currently does not consider elements yet to be implemented. We consider that having means of reporting the coverage of the ontology can be achieved with TDD.

DL queries are ”Test After” so not apt for test driven development. According to Malone et al. [8], at the time of the publication of the Software Ontology (SWO) the testing infrastructure for developing an ontology with a test-driven approach was not mature enough. We consider that this is no longer the case. Using a multi-platform containerized software, one could set up an infrastructure that loads the ontology into a graph database like a Triplestore or a Labelled Property Graph (LPG). These would be fed using dummy data to perform ”Test Before” operations in the form of queries. These queries then could be paired with expected outputs. After this one should have a collection of failing tests that would pass once the associated features are implemented.

The difficult step is turning competency questions into queries. According to Wiśniewski et al. [9] there is no one-fits-all way of doing this. One reason is the particular formulation of questions. Another one is the design limitations of the ontologies themselves. Since we are focusing on energy systems data, we attempt to mitigate these concerns by writing competency questions using a limited scope.

1.3. Collecting competency questions

There is no established process to produce competency questions of an ontology since these are specific for each application. To encourage deeper discussion, a series of theoretical approaches is outlined. One approach are interviews with people who work with models. For example, Cao et al. [10] produced a transparency checklist this way. This approach considers concerns that are in line with the day-to-day use of models by scientists. The quality of this method depends a lot on the execution, since it deals directly with people with different scientific backgrounds. Besides, the medium used for the interviews is also crucial, since the outcomes of a survey may be very different from those of a workshop. The latter allows collaborative work which can be

very productive, as shown by the development of the Software Ontology (Malone et al. [8]).

Another approach is a classical meta-analysis of the existing literature. There are already publications that deal with transparency and reproducibility in the field. Some examples are the previously mentioned checklist from Cao et al. [10], the data harmonization tool from Gotzens et al. [11] and the model comparison from Gils et al. [12]. The competency questions could be formulated using the definitions and approaches defined in these papers. This has the advantage of being grounded in existing literature. Unlike the previous example, a practical problem is that the methods in this field evolve fast and there could be a temporal gap between the reports and what is being done right now.

2. Methodology

To demonstrate on a small example how formulating competency questions works, we will do a compilation consisting of the curation, formulation and translation of questions based on existing energy research. This demonstration has three groups, each covering a different aspect of energy systems analysis and containing three example questions. The first group relates to documentation and transparency. These are adapted from Cao et al. [10] transparency questions. The second group consists of one example of data provenance and processing, also using Cao et al. [10] and Gotzens et al. [11]. The last group consists of a compilation of questions based on the structured multimodel output analysis performed in Henry et al. [4] and Gils et al. [12].

Inspired by Wiśniewski et al. [9], the questions are written with querying languages in mind as to allow further research using both graph databases and semantic web software. For this task we use SPARQL as an example of a language used in the context of linked data. The data model on which SPARQL queries are performed uses what is called the Resource Description Framework (RDF) as basis. We use Cypher as an example of a more practical enterprise-focused querying language. It is important to note that the LPG model, on which Cypher queries are performed, is not very strict concerning the entities used in it. As of today, a data engineer has to use neo4j with its extension called neosemantics (Jesús Barrasa [13]) to include semantics in an LPG. The reasoning behind each one of the query construction exercises is documented to be an example for the production of similar data.

3. Results

The resulting competency questions are grouped in three groups. The first group (1) contains questions about transparency, the second about data sharing and discovery (2) and the third about analysis and comparison (3). Each question is then assigned a letter, for example, the first question gets assigned the label 1.A. The queries can be found in the Table 1. This is a detailed explanation of the reasoning behind the query construction.

Table 1: Competency question queries, left queries are in Cypher and right queries are in SPARQL. The queries use the prefix ont/ONT to refer to an arbitrary ontology and rdfs to the RDF schema. The <id> placeholder is used in both cases to refer an identifier in a database.

1.A: Who are the authors of the study and for which institutions do they work?

```
MATCH (study:ONT_STUDY {id:
 ↪  "<id>"})-[:HAS_AUTHOR]->

 ↪  (author:ONT_AUTHOR)-[:WORKS_AT]->
      (institution:ONT_INSTITUTION)
RETURN author.name, institution.name
```

```
SELECT ?author ?institution
WHERE {
    ?author ^ont:has_author ?study;
            ont:works_at ?institution .
    ?study rdf:type ont:study;
           ont:has_identifier <id> .
      }
```

1.B: ”What is the URL to the source code of the used model(s)?

```
MATCH (study:ONT_STUDY {id:
 ↪  "<id>"})-[:USES_TOOL]->

 ↪  (model::ONT_ENERGY_SYSTEMS_MODEL)-
 ↪  [:HAS_REPOSITORY]->
      (repository:ONT_REPO)
RETURN model.name, repository.url
```

```
SELECT ?model ?url
WHERE {
?repository ^ont:has_repo ?model;
            ont:has_url ?url .
    ?model ^ont:uses_tool ?study;
        rdf:type ?esm .
    ?study rdf:type ont:study;
        ont:has_identifier <id> . }
```

2.A: What is the installed electrical capacity of Germany?

```
MATCH (plant:ONT_PLANT)-
 ↪  [:HAS_COUNTRY]->
      (country:ONT_COUNTRY)
WHERE country.name == "Germany"
RETURN country.name,
 ↪  sum(plant.capacity)
```

```
SELECT ?plant ?country ?capacity
WHERE
{ ?plant rdf:type ont:PowerStation
OPTIONAL { ?plant ont:country ?country }
OPTIONAL { ?plant
 ↪  ont:psElectricalCapacity ?capacity}
FILTER (regex(?country, "Germany", "i"))
 ↪  }
```

2.B: Which study cost values with reference and publication years are in the database?

```
MATCH (study:ONT_STUDY {about:
↪  "ONT:PHOTOVOLTAIC"})-[:HAS_VALUE]-
↪  >
      (capex:ONT_INVESTMENT_COST)
RETURN capex.value, capex.units,
↪  study.doi, study.pub_year,
↪  capex.ref_year
```

```
SELECT  ?value ?units ?study ?pub_year
 ↪  ?ref_year
WHERE { ?value ^ont:has_value ?study ;
        rdf:type ont:investment_cost;
        ont:has_units ?units .
        OPTIONAL { ?value ont:ref_year
         ↪  ?ref_year . }
SELECT ?study, ?pub_year
WHERE { ?study rdf:type ont:study ;
    ont:is_about ont:Photovoltaic ;
    ont:year_published ?pub_year . } }
```

3.A: What are the system costs estimated by the energy systems models(ESM) in the study?

```
MATCH (study:ONT_STUDY {id:
↪  "<id>"})-[:USES_TOOL]->
      (model:ONT_ESM)-[:HAS_VALUE]->
      (cost:ONT_SYSTEM_COST)
RETURN cost.magnitude, cost.untis,
↪  model.name
```

```
SELECT  ?value ?units ?model
WHERE {
?value ^ont:has_value ?study ;
        rdf:type ont:system_cost;
        ont:has_source ?model ;
        ont:has_units ?units .
?model rdf:type ont:esm .
?study rdf:type ont:study;
        ont:has_identifier <id> . }
```

3.B: What are the model system costs in relation to CO_2 limits?

```
MATCH (study:ONT_STUDY {id:
↪  "<id>"})<-[:PART_OF]-
      (output:ONT_MODEL_OUTPUT)<-
↪  [:HAS_SOURCE]-

↪  (model:ONT_ENERGY_SYSTEMS_MODEL)),
      (output)-[:HAS_VALUE]-
↪  >(cost:ONT_SYSTEM_COST),
      (output)-[:HAS_VALUE]-
↪  >(co2_limit:ONT_CO2_LIMIT)
RETURN cost.magnitude cost.units
↪  co2_limit.magnitude
↪  co2_limit.units, model.name
```

```
SELECT  ?cost ?cost_u ?co2_limit ?limit_u
 ↪  ?model
WHERE
{ ?cost ^ont:has_value ?model_output ;
        rdf:type ont:system_cost;
        ont:has_units ?cost_u .
?co2_limit ^ont:has_value ?model_output ;
        rdf:type ont:co2_limit ;
        ont:has_units ?limit_u .
SELECT ?model_output, ?model
WHERE { ?study rdf:type ont:study;
    ont:has_identifier <id> .
?model_output ^ont:part_of ?study;
              ont:has_source ?model;
              rdf:type ont:model_output
               ↪  .}}
```

To produce 1.A ”Who are the authors of the study and for which institutions do they work?” the question ”Who are the ESS authors and for which institution(s) do they work?” from Cao et al. [10] was used as a basis. The concept ESS refers to ”model-based energy scenario study” in the original publication, we simplify the term to ”study” because in the case of this particular question it is not necessary to define a special entity for the ESS since any kind of study can have their respective authors. The study is referred to via a unique identifier <id>. The query first searches for the study based on the identifier. Then it searches for all the authors associated with the study using the ”has_author” relationship with a reverse operator ”^”. From those authors, we get the ”works_at” relationship, whose domain should be an institution. Both the author and institution are returned as output. In contrast, the Cypher query is way more compact. In this case, the id is contained within the node information and it is connected to the author using the ”HAS_AUTHOR” relationship which in turn is connected to its institution with the ”WORKS_AT” relationship.

The question 1.B is based on: ”Is the model’s source code accessible?”. The question was reformulated to ””What is the URL to the source code of the used model(s)?”. It emphasizes the usage of multiple models. The SPARQL query will return an empty response if any of the models does not have a URL. This can be avoided if the OPTIONAL clause is used. The Cypher query is an example where information of the entity is stored within the node with the WHERE clause checking for the model type and the URL included as part of the ”repository” node contents.

Searching for data from earlier research is a common activity in energy systems analysis. Gotzens et al. [11] states that power plant data can be inconsistent across sources and provenance documentation is important for its transparency. 2.A ”What is the installed electrical capacity of Germany?” is a simple example of how power plant information could be extracted from a knowledge graph. This query is directly executable in the DBPedia (Auer et al. [14]) SPARQL endpoint if one replaces the generic ”ont” with the DBPedia ontologies (prefix dbo in the site, dbp for the electrical capacity). The query searches for all objects mapped to the type PowerStation. The OPTIONAL clause ensures that the entries without capacity or country are also considered, the FILTER clause ensures we get values only for entries with Germany in the country relationship. The Cypher query is written assuming that the capacity of the power plant is information inside the node. If these queries were to be further differentiated by source, an evaluation exercise very similar to the analysis section in Gotzens et al. [11] could be done.

Question 2.B is a typical example for a request for techno-economic data: ”Which study cost (capital investment) values with reference and publication years are in the database?”. Here, we differentiate between publication year and reference year. Reference years are a very important part of techno-economic data reports as currency reference values vary with time. There is, however, the risk of a report not stating explicitly its reference year. That is why we use the OPTIONAL clause in the SPARQL query.

Both Henry et al. [4] and Gils et al. [12] perform a comparison of the outputs of different optimization models. To reproduce the tasks outlined in these publications in a generalized way one has to include the annotated outputs of the models in a database and perform queries like 3.A ”What are the system costs estimated by the energy systems models in the study?”. The SPARQL query considers that there are multiple values in a single study. These values have each a source associated to one model. This is used in contrast to having a monolithic value

container for a model output. This latter option is explored in 3.B. In the Cypher query, we assume that the values would be part of the nodes themselves. This allows for a compact query at the price of having ontology terms as fields which can hinder semantic search.

The question 3.B "What are the model system costs in relation to CO_2 limits?" takes advantage of having individual model data bundled together in a model output entity. In this case the "has_value" relationship should be connected to a different entity called model output which in place would be connected to a study with a relationship such as "part_of".

4. Discussion

According to our research there is more than one way to construct the queries as in section 3. The main difference of these alternative structures is that one or more of the entities mapped to a container could be instead mapped to an overarching entity. For example, in 1.B one could map the URL property directly to the model entity without a major loss of information. However, one drawback would be that the repository entity would be lost and thus the opportunity of attaching multiple URLs to the model associated to its other possible web addresses like its documentation. The structure of the LPG allows arbitrary allocation of string and number values. The way in which an engine processes this node information is different to how it processes the information of its relationships (Robinson et al. [15]). This has implications for computing performance. The RDF store model is more rigid on how to include string and number information. The performance of these queries depends on what technology is used to host the data.

In all the cases, it was noted that there was a need of expertise. The construction of the queries requires not only a natural language question but also a deep understanding of the concepts in them. If this knowledge is disregarded one could fall into creating queries with misconceptions embedded. For example, in 2.A the capacity of a power plant can have different meanings, depending on the boundaries used to estimate it. Nominal capacity is not the same as the actual output of the plant in a given year. It is up to the person reading a report to tell the difference.

When designing the structure of a knowledge graph, two decisions have to be made. The first is ease of use for human users. How complicated is it to write new queries? The second is about performance. Do queries need a lot of computational resources? These aspects are beyond the scope of this paper but are not to be ignored during graph design decision-making.

These queries for demonstration are certainly not optimal in terms of performance. We see a huge potential to build up large data sets of such queries. It is important to note that there is no one-size-fits-all solution concerning ontologies and competency questions. Each ontology is adapted to its particular field and this also applies to competency questions. If questions are too general, compromises in readability, searchability and computing have to be made. From this exercise we propose that in order to create valuable queries, smaller expert groups get together to tackle a well delimited topic within the field. This said, we think that the role of the knowledge engineer is not to build the competency questions all by herself but to orchestrate activities that will produce them.

5. Conclusion

We consider that ontologies increase transparency and reproducibility in many scientific domains. In energy systems research, ontologies like the OEO only now begin to emerge as tools. In this paper we propose several methods of gathering research questions and convert them into competency question queries. Using one of these methods, we constructed nine queries in two languages associated to three aspects of energy systems research. We found out that most of the effort of writing such queries goes into the abstraction of natural language into formal semantics. We found that using Cypher one can build more compact and user-friendly queries whereas SPARQL allows for stricter semantic search. We also noted that there is several ways in which similar information can be conveyed, each has advantages and compromises. Furthermore, we propose that given the level of expertise required to build one of these questions, this exercise can be more effective when done by smaller groups of experts with knowledge about a particular topic of the field. These queries may serve as starting points for testing ontologies in a knowledge graph implementation context. In our case, we expect to integrate and demonstrate our efforts within the context of the OEO.

Acknowledgments

The research for this paper was performed in the framework of the LOD GEOSS project supported by the German Federal Ministry of Economic Affairs and Climate Action (BMWK) under grant number 03EI1005A.

References

[1] X. Chen, S. Jia, Y. Xiang, A review: Knowledge reasoning over knowledge graph, Expert Systems with Applications 141 (2020) 112948. doi:10.1016/j.eswa.2019.112948.

[2] S. Pfenninger, A. Hawkes, J. Keirstead, Energy systems modeling for twenty-first century energy challenges, Renewable and Sustainable Energy Reviews 33 (2014) 74–86. doi:10.1016/j.rser.2014.02.003.

[3] J. DeCarolis, H. Daly, P. Dodds, I. Keppo, F. Li, W. McDowall, S. Pye, N. Strachan, E. Trutnevyte, W. Usher, M. Winning, S. Yeh, M. Zeyringer, Formalizing best practice for energy system optimization modelling, Applied Energy 194 (2017) 184–198. doi:10.1016/j.apenergy.2017.03.001.

[4] C. L. Henry, H. Eshraghi, O. Lugovoy, M. B. Waite, J. F. DeCarolis, D. J. Farnham, T. H. Ruggles, R. A. Peer, Y. Wu, A. de Queiroz, V. Potashnikov, V. Modi, K. Caldeira, Promoting reproducibility and increased collaboration in electric sector capacity expansion models with community benchmarking and intercomparison efforts, Applied Energy 304 (2021) 117745. doi:10.1016/j.apenergy.2021.117745.

[5] M. Booshehri, L. Emele, S. Flügel, H. Förster, J. Frey, U. Frey, M. Glauer, J. Hastings, C. Hofmann, C. Hoyer-Klick, L. Hülk, A. Kleinau, K. Knosala, L. Kotzur, P. Kuckertz, T. Mossakowski, C. Muschner, F. Neuhaus, M. Pehl, M. Robinius, V. Sehn, M. Stappel, Introducing the open energy ontology: Enhancing data interpretation and interfacing

in energy systems analysis, Energy and AI 5 (2021) 100074. doi:10.1016/j.egyai.2021.100074.

[6] L. Madeyski, Test-Driven Development, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. doi:10.1007/978-3-642-04288-1.

[7] M. Gruninger, Methodology for the design and evaluation of ontologies, undefined (1995).

[8] J. Malone, A. Brown, A. L. Lister, J. Ison, D. Hull, H. Parkinson, R. Stevens, The software ontology (swo): a resource for reproducibility in biomedical data analysis, curation and digital preservation, Journal of biomedical semantics 5 (2014) 25. doi:10.1186/2041-1480-5-25.

[9] D. Wiśniewski, J. Potoniec, A. Ławrynowicz, C. M. Keet, Analysis of ontology competency questions and their formalizations in sparql-owl, Journal of Web Semantics 59 (2019) 100534. doi:10.1016/j.websem.2019.100534.

[10] K.-K. Cao, F. Cebulla, J. J. Gómez Vilchez, B. Mousavi, S. Prehofer, Raising awareness in model-based energy scenario studies—a transparency checklist, Energy, Sustainability and Society 6 (2016). doi:10.1186/s13705-016-0090-z.

[11] F. Gotzens, H. Heinrichs, J. Hörsch, F. Hofmann, Performing energy modelling exercises in a transparent way - the issue of data quality in power plant databases, Energy Strategy Reviews 23 (2019) 1–12. doi:10.1016/j.esr.2018.11.004.

[12] H. C. Gils, H. Gardian, M. Kittel, W.-P. Schill, A. Murmann, J. Launer, F. Gaumnitz, J. van Ouwerkerk, J. Mikurda, L. Torralba-Díaz, Model-related outcome differences in power system models with sector coupling—quantification and drivers, Renewable and Sustainable Energy Reviews 159 (2022) 112177. doi:10.1016/j.rser.2022.112177.

[13] Jesús Barrasa, neosemantics (n10s), 2022. URL: https://github.com/neo4j-labs/neosemantics.git.

[14] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: A nucleus for a web of open data, in: Proceedings of the 6th International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference, ISWC'07/ASWC'07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 722–735.

[15] I. Robinson, J. Webber, E. Eifrem, Graph databases: Compliments of Neo Technology, 1. ed. ed., O'Reilly, Beijing and Köln, 2013.