Use of Natural Language Processing in Digital Engineering Context to Aid Tagging of Model

Daniel Dunbar
School of Systems and Enterprises
Stevens Institute of Technology
Hoboken, NJ
0000-0001-9647-8073

Maximilian Vierlboeck
School of Systems and Enterprises
Stevens Institute of Technology
Hoboken, NJ
0000-0002-9518-1216

Mark Blackburn
School of Systems and Enterprises
Stevens Institute of Technology
Hoboken, NJ
mblackbu@stevens.edu

***Abstract*—This paper uses Natural Language Processing to provide augmented intelligence assistance to the resource intensive task of aligning systems engineering artifacts, namely text requirements and system models, with ontologies. Ontologies are a key enabling technology for digital, multidisciplinary interoperability. The approach presented in this paper combines the efficiency of statistical based natural language processing to process large sets of data with expert verification of output to enable accurate alignment to ontologies in a time efficient manner. It applies this approach to an example from the telecommunications domain to demonstrate the workflows and highlight key points in the process. Enabling easier, faster alignment of systems engineering artifacts with ontologies allows for a holistic view of a system under design and enables interoperability between tools and domains.**

Keywords—ontology, natural language processing, semantic web, digital engineering, authoritative source of truth, augmented intelligence

I. INTRODUCTION

Digital interoperability between different data models is becoming increasingly important. The growing complexity of engineered systems and the interrelated nature of many aspects of a system means that common understanding across models will continue to grow in importance.

Ontologies provide a common communication and discussion vocabulary for different tools and models to use when talking with each other, which makes them great theoretical tools for interoperability [1]. However, terms within ontologies must be explicitly aligned to terms used within disparate models in order to achieve the stated interoperability potential of ontologies. This alignment process is often called mapping, and while it can be effective, it can also be time and resource intensive.

This paper presents an approach to enhance the ability to quickly and accurately map data from requirements and system models to ontologies. Combining Natural Language Processing and expert verification enables an augmented intelligence approach for this alignment process and applies existing research into the parsing of requirements and the tagging of system models. By increasing the efficiency with which these system artifacts can be aligned to ontologies, this paper seeks to bridge the gap from theoretically useful to practically feasible.

II. BACKGROUND AND STATE OF THE ART

A. Ontologies in Systems Engineering

Systems themselves can be captured in terms of ontologies, and benefits can be derived from such a form of knowledge representation. Hennig et al. [2] use automated reasoning enabled by an ontological understanding of a system to perform tasks such as engineering discipline allocation to certain system elements and creation of Critical Item Lists based on axioms defined in the underlying ontologies. Dunbar et al. [3] demonstrate the ability to perform verification tasks on ontology-aligned data to determine completeness of a model according to context specific definitions. In another paper, Dunbar et al. [4] demonstrates how ontology-aligned data can be used to enable Digital Thread applications.

In order to perform some of these analyses on systems represented in ontologies, mechanisms for transforming a system from more common representations, such as the Systems Modeling Language (SysML), to ontology-aligned data must be established. At NASA's Jet Propulsion Lab (JPL), research has been performed that uses the SysML stereotype as a mechanism for tagging a SysML model with terms corresponding to classes in ontologies [5], [6]. Dunbar et al. [4] introduce a new interface that allows external tools to read and write to ontology-aligned data without going through a mapping process. However, this new interface is geared more towards model based engineering tools outside of systems engineering, and the SysML model transformation also uses stereotypes as tags for a mapping process to the ontology-aligned data.

While the use of custom SysML stereotypes as tags is beneficial for explicitly tying SysML system models to ontological classes, it can quickly become unwieldy as the models continue to grow. Manual tagging of tens or a couple of hundred elements may be arduous, but it is doable. Manual tagging of thousands of elements is an order of magnitude higher and difficult if not impossible to achieve.

B. Natural Language Processing for Requirements Engineering

Natural Language Processing (NLP), despite some controversial definitions for it [7], can be defined in its core as the attempt to process natural language with computer tools that are supposed to allow a human-like linguistic analysis and manipulation of text/speech [7]–[9]. This definition is very broad and covers a plethora of possible applications. This diversity is also seen in the multitude of research directions that emerged over time for NLP. Since the topic of this publication concerns ontologies and requirements in particular, the applicable research and approaches are fortunately more specific.

In general, NLP in conjunction with ontologies and requirements is targeting what is called developed text analysis. For this type of analysis, three categories exist [10]: syntactic, semantic, and lexical methods. The first, syntactic approaches, are concerned with the structure of sentences and the grammatical constructs therein. The second, semantic approaches, address the logical structure of a sentence. While robust, semantic approaches require a set sentence structure, without which they cannot properly function. Lastly, lexical methods can be considered the most robust [10] due to the fact that they do not rely on part-of-speech analysis, for example. Instead, lexical approaches work on the level of the character sequence to analyze the text.

Specifically for requirements, a whole research field exists called Natural Language Processing for Requirements Engineering (NLP4RE). A comprehensive overview for the NLP4RE field was provided recently by Zhao et al. [11], who compared a vast number of approaches in the NLP4RE sector. Based on said review, and thus also associated with the field is the approach that is applied in the presented research, shown by Vierlboeck, Nilchiani, and Blackburn [12]. The applied syntactic approach allows for the decomposition of requirements into structural networks that contain not only the connections between the requirements, but also the specific elements therein. The result of this approach is a structured body of information that represents the requirements as well as their connections and links. These results and the content of the approach, enabled through NLP, form a vital part of the concept presented in this paper.

C. *Similarity Metrics*

When it comes to mapping and tagging, identifying the same terms, similar terms, and/or related terms is crucial. Yet, when it comes to natural language, not all related terms are equally similar. Thus, defining similarity requires the assessment of potential approaches for classifications and/or clustering to find connections.

One, and maybe the most prominent field that addresses the problem above, is distributional semantics [13]. Distributional semantics targets the meanings of words and defines their interpretation in different contexts. As a result, words that occur in the same context are considered as similar in meaning [13] and thus, a classification is enabled. While not the only approach to implement such techniques, the most popular one is the use of vectorization. Vectorization allows for the algebraic definition of identifiers and as a result, the similarity or closeness of terms can be calculated [14]. These calculations allow for clustering and classification.

Another classification/clustering approach, which in part is related to the context use in word embeddings, is the use of similarity measures such as syntactical, contextual, and lexical similarity, as outlined by Nenadić et al. [15]. Using such measures (either individually or as a combination [16]) can allow for the definition of similarity and thus grouping/clustering. This clustering can then be used for taxonomy building.

In addition to the approaches above, simpler approaches based on character similarity and root words have to be mentioned since they, while not necessarily targeting the meaning and semantics, can be essential for the ontology mapping and organization. For instance, root words can be used to connect word families as well as tenses for verbs and lemmatization in general.

When it comes to the problem at hand specifically, another issue to keep in mind is the circumstantial factor of domain specificity. Since ontologies and the NLP approaches discussed in this paper have certain domain dependent elements, the context and overall domain has to be incorporated accordingly. The issues and difficulties concerning domain specificity and interpretational contexts have been evaluated by Lipizzi et al. as well [17], [18].

All in all, the literature and information above shows that there are different ways to address the issues discussed in this paper regarding ontology processing as also discussed by Kof [16]. How the information above is used in the concept is outlined in the following sections.

While work has been performed in ontology-aligned data in systems engineering, NLP4RE, and similarity metrics, a combination of the fields provides a unique contribution to the Systems community. In the combination, an opportunity for augmented intelligence emerges that allows various NLP algorithms, combined with closeness metrics comparing NLP output to various domain ontology classes, to aid in the tagging of a system model. This brings alignment between three previously separate bodies of knowledge within a system design – the requirements used to inform, verify, and validate the design, the system model that houses the architecture and an interdisciplinary view of the system under design, and domain ontologies that provide a formal vocabulary for describing the domain contexts for the design and provide capacity for the transformation of the design into a graph data structure.

III. METHODS

The process of using NLP to assist in the alignment process can be broken into two separate workflows: the alignment between requirements and ontologies and the alignment between a system model and ontologies.

A. *Requirements and Ontologies Alignment Workflow*

With requirements, the end goal of an ontology alignment workflow is to identify all terms within the requirement set that correspond to existing or potential ontology classes within the given set of used ontologies. This paper uses a workflow to achieve this that combines NLP and Expert Verification (Fig. 1).

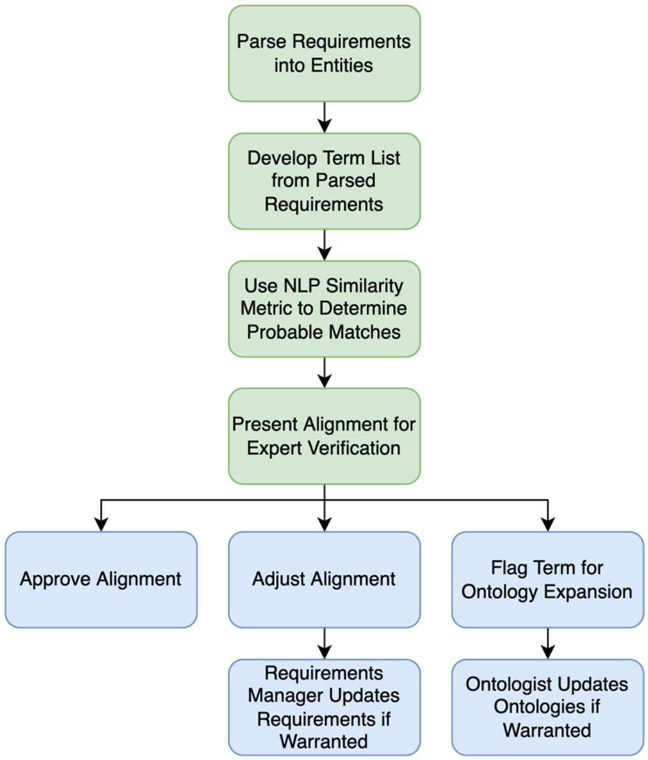

Fig. 1. Alignment Between Requirements and Ontologies

The first part of the process uses NLP to parse text requirements into relevant entities similar to the technique described by Vierlboeck, Nilchiani, and Blackburn [12]. From this output a term list can be compiled to be used in a similarity analysis. As described in the Background section of this paper, there are various approaches to assessing similarity of words. Further research can be done to determine what approach works best in this context, but the researchers recommend structuring the workflow in a way that allows the NLP similarity analysis to be a modular component of a larger workflow. Similarity analysis, from the workflow perspective, can be seen as a black box where two lists of terms are used as inputs: the term list from the requirements and the list of classes in the used ontologies. The outputs are ordered lists of possible matches from the second list (ontologies) that correspond to each term in the first list (requirements) along with a score of each possible match. Given the modularity of the similarity analysis, different forms of analysis can be used and modified to best fit the given context.

An essential part of the workflow is expert verification. Given the precise nature of engineering projects, the formal and unambiguous nature of ontologies, and the interoperability functionality that ontologies are meant to play in this context, a stochastic approach to alignment of requirements or system model elements to ontological classes in insufficient by itself. However, when combined with expert verification, stochastic approaches and NLP provide immense value by processing data well and presenting it for human verification in a way that limits the stochastic nature of the approach. For requirements, the potential matches can be presented to an expert in a way that allows the expert to process the matches rapidly. There are three possible outcomes of the verification:

1. Approve Alignment – in this case, the expert verifies that the NLP algorithm chose the correct class to correspond with the requirement (or system element in the next workflow).
2. Adjust Alignment – in this case, the expert sees that the NLP algorithm did not choose the correct class and corrects the alignment. It may be that the correct option was further down the list of possible classes, or it may be that the NLP algorithm did not see the correct class as a possibility. It also may be that in this process the requirement (or system element in the next workflow) is misspelled or otherwise incorrect in the original source material, and a correction can be made.
3. Flag Term for Ontology Expansion – in this case, the expert sees a valid term in the requirement (or system model), but there is no ontology class that corresponds to the element. This is an indication that an ontologist needs to expand the used ontologies to capture the new term. In this way, the expert verification process can actually aid in the expansion of ontologies that can then be reused in future projects.

This workflow aims to align requirement terms with ontology classes. However, there will certainly be terms included in most requirement sets that do not correspond to classes within an ontology. For example, the requirement "The transmit antenna shall have a gain of no less than 9 dB" includes some elements that would correspond to an ontological class – transmit antenna and gain are both concepts that could be captured in an ontology. However, "9 dB" is a descriptor of the gain of the antenna, and it would be considered an instance of gain and not correspond to an ontological class itself. Likewise, the ontologies used in a project will almost certainly contain classes that have no match in the requirement set. Perhaps the ontology is being reused from a previous project that has slightly different specifics, so some classes were developed that will not be used in this project. Thus, while alignment is sought between the requirements and the ontologies, 100 percent alignment is not the goal of the workflow (Fig 2).

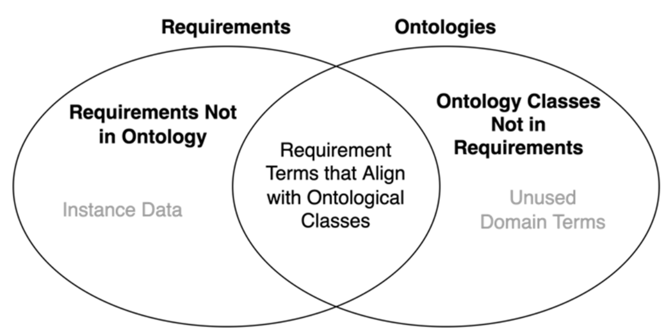

Fig. 2. Venn Diagram Showing the Overlap Between Requirements and Ontologies

B. *System Model and Ontologies Alignment Workflow*

With a system model, the end goal of an ontology alignment workflow is to identify all elements within a system model that correspond to existing or potential ontology classes within the given set of used ontologies and to tag the system model with the relevant ontological class term. This goal is similar to the requirements alignment goal, and the workflow (Fig 3) is similar as well, but it has some key differences.

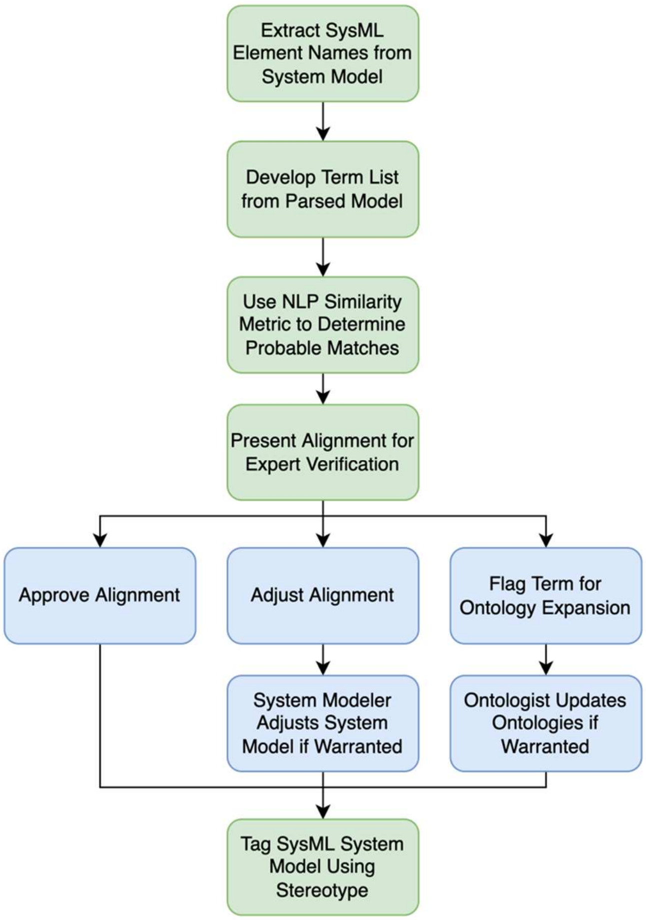

Fig. 3. Alignment Between SysML System Models and Ontologies

Similar to the requirements workflow, this workflow begins with the development of a terms list. However, this workflow does not need NLP to extract terms. Since SysML element names are typically short and descriptive, they can be extracted as is. After this, the same similarity metric used in the requirements workflow can be used to provide probable matches between the developed term list and the classes present in the used ontologies. Expert verification is then performed on the probable matches in a similar manner to the requirements workflow.

A significant difference in this workflow exists after alignment has been performed. In the requirements workflow, it is reasonable to have direct alignment when the ontology and the requirement are discussing the same concept. Discrepancies can thus be addressed by updating the requirement language to align with the ontological term in the ontology. In a system model, there are other reasons for naming an element something different than the ontological class associated with that element. For example, in an aircraft system model, a block may be named “Left Rear LG,” and its tag may be “Landing Gear Structure." The block naming is interesting from an architectural standpoint, and the modeler may insist on keeping it instead of “correcting” it to align with the ontology term. However, the tag (eventually instantiated by a stereotype in the SysML model) will need to align directly with the ontology term. Thus, while the element name is used in the similarity metric to provide possible matches to an ontology, the alignment is actually performed between an added stereotype in the system model and the ontology classes.

IV. Results and Validation

The shown approach has been applied to a conceptual model from the telecommunication domain [19]. The test case system is a two-way radio communication system, and the particular design task is centered around a tower site that has a tower subsystem and a shelter subsystem, which houses most of the radio equipment for the site. A basic ontology has been developed to capture various terms associated with the telecommunications system. Table I shows a partial list on ontology classes that are part of the conceptual use case.

TABLE I. Partial List of Ontology Classes Related to Telecommunications Domain

Radio Access Network	RAN Tx Antenna Type	Antenna Gain
RAN Tower Site	RAN Tx Antenna Height	FCC ASR Number
RAN Tower Subsystem	RAN Tx Mainline Type	RAN Transmitter Power
RAN Shelter Subsystem	RAN Tx Mainline Length	RAN Frequency Band

A. Requirements Alignment

Requirements are developed in the Systems Engineering process to provide design criteria and constraints necessary for successful implementation of the system. Consider the three example textual requirements from Table II.

TABLE II. Requirements Table

Req Num	Requirement Text
1	The Transmit Antenna shall provide a minimum of 9 dB of gain.
2	The RAN Tower shall provide a minimum of 92% Service Area Covered.
3	The frequency band used by the RAN equipment shall use the 700 MHz Public Safety band.

Following the first two steps in Fig. 1, the text requirements can be parsed into separate entities according to the methodology set forth by Vierlboeck et al. [18]. This parsing can then be used to form a term list (Table III).

TABLE III. TERM LIST FROM PARSED TEXT REQUIREMENTS

Req Num	Term List		
1	Transmit Antenna	Gain	
2	RAN Tower	Service Area Coverage	
3	Frequency Band	RAN equipment	Public Safety band

Notice that some key information included in the textual requirements is left out of this term list. As discussed in the Methods section, the specification of 9 dB of gain in the requirement gives both a class of data (antenna gain) as well as an instance of the class that contains the constraint (9 dB). Thus, while the mapping process discussed below seeks to align the classes with the requirements, additional work will need to be done to create instances that provide additional constraints.

The requirement term list can serve as an input to an NLP similarity module, along with the ontology class list from Table I. While several approaches to similarity between texts exist in NLP literature, some of which were discussed in previous sections of this paper, this conceptual use case refrains from choosing any particular similarity metric for demonstration purposes. The NLP similarity metric can be considered a black box with regards to the rest of the activity flow, and different domains or users may find that different similarity metrics work better for their contexts. What is important is that the NLP module take two inputs (ontology class list and requirements term list) and provide matching, preferably with a degree of confidence about the match to help guide expert verification in the next step.

With similarity established, the preliminary matching can be presented to an expert for verification and modification. In Fig. 4, a mockup of a dashboard can be seen that shows the requirement terms, along with a reference to what requirement number the term is parsed from for easy reference. Next to the requirement term is an ontological class, and the dashboard uses colors to indicate match quality. Green indicates an exact match between requirement term and ontology class, orange indicates a likely match that isn't exact, no fill represents an unknown or weak match, and blue represents a user verified or modified match. While spacing in this paper does not allow, additional fields could be put in this mockup for space to write notes to requirements engineering and/or the ontology group. This would be useful in situations where a flaw in the requirement is found during the mapping process or an additional ontology class is needed to complete the mapping process.

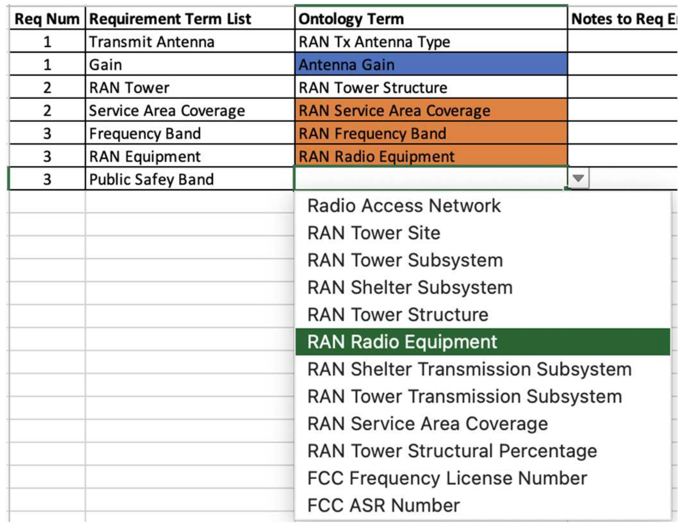

Fig. 4. Mockup Dashboard of a Requirements-Ontology Alignment Process

B. *SysML Model Alignment*

Fig. 5 shows a SysML model of a simplified architecture for a tower site associated with a Radio Access Network (RAN). This architecture is untagged, but in order to use it with the Digital Engineering Framework for Integration and Interoperability (DEFII) mentioned in Dunbar et al. [4], tags must be assigned to explicitly declare what ontology classes are represented in the SysML model. This paper will use the SysML stereotype to provide the tagging functionality.

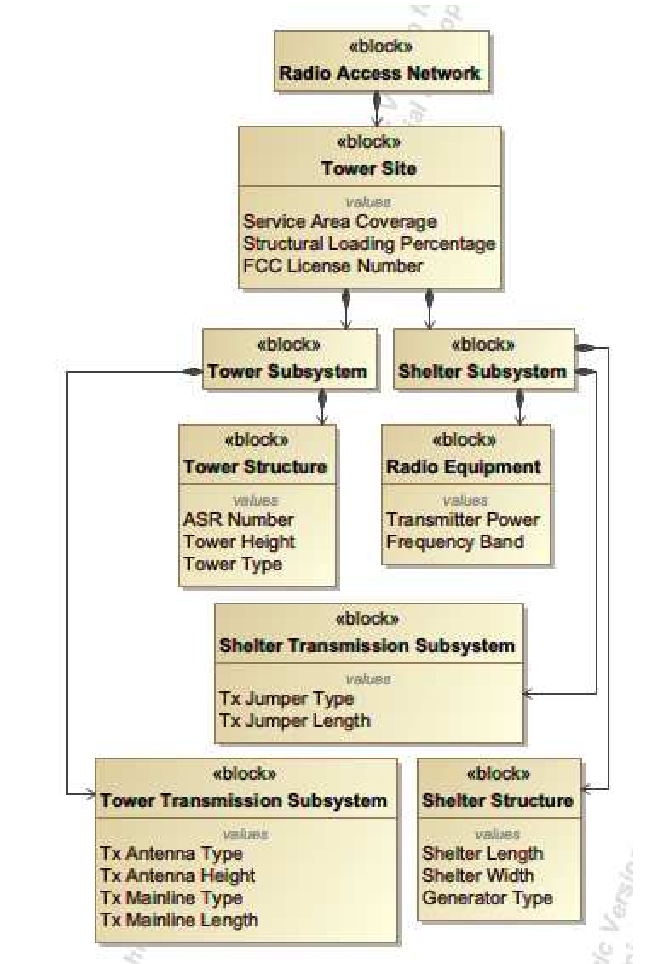

Fig. 5. SysML Block Definition Diagram of a Simplified Telecommunications Radio Access Network

Names associated with blocks and value properties in the diagram can be pulled to form a term list for the SysML model (Table IV).

TABLE IV. PARTIAL SYSML TERM LIST

Element ID	SysML Terms
bad7g	Radio Access Network
kqwb8	Tower Site
jaa7v	Tower Subsystem
xn1h2	Shelter Subsystem
ppaj2	Tower Structure
vnah3	Tower Transmission Subsystem
11bvh	Radio Equipment

Similar to the requirements alignment, this term list is used as an input alongside the ontology class list to an NLP module to provide a similarity metric and mapping suggesting between the SysML model and the ontology. Fig. 6 shows a mockup of the Augmented Intelligence dashboard that aids an expert in the verification process of the mapping between the ontology and the SysML model. As shown above, the coloring of the ontology cells provides additional information to the expert to assist in the mapping process.

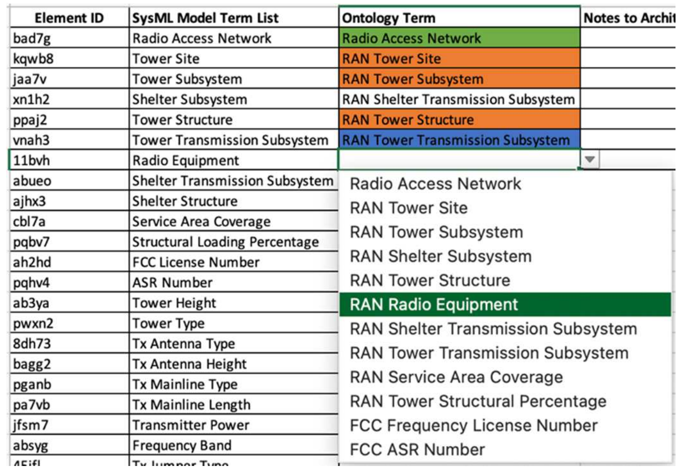

Fig. 6. Mockup Dashboard of a SysML Model-Ontology Alignment Process

Note in Fig. 6 that the “Shelter Subsystem” SysML item has been incorrectly matched with the “RAN Shelter Transmission Subsystem” ontology class. Part of the expert verification step is correction of improper matching. This step is necessary and should not be overlooked with the assumption that the NLP matching algorithm is infallible. The proposed method here assumes limitations in the NLP algorithm and deficiencies in inputs (ontologies, requirements, and system models), so this verification step is an active check on the process. This form of augmented intelligence enables acceleration of workflow, but it does not claim to be autonomous, nor does it need to in order to provide clear value to the users.

While the completion of the alignment process may be sufficient for the requirements portion of this paper, it is useful to add the additional step of adding the custom SysML stereotypes corresponding to ontology classes to the appropriate elements within the SysML model. In the DEFII framework, the Authoritative Source of Truth (AST) is the ontology-aligned data, so if the data has been fully mapped, the stereotype could theoretically be left out. However, for clarity as well as ease of mapping in the future when changes occur in the architecture, it is good practice to update the SysML model with the custom stereotypes, as seen in Fig. 7.

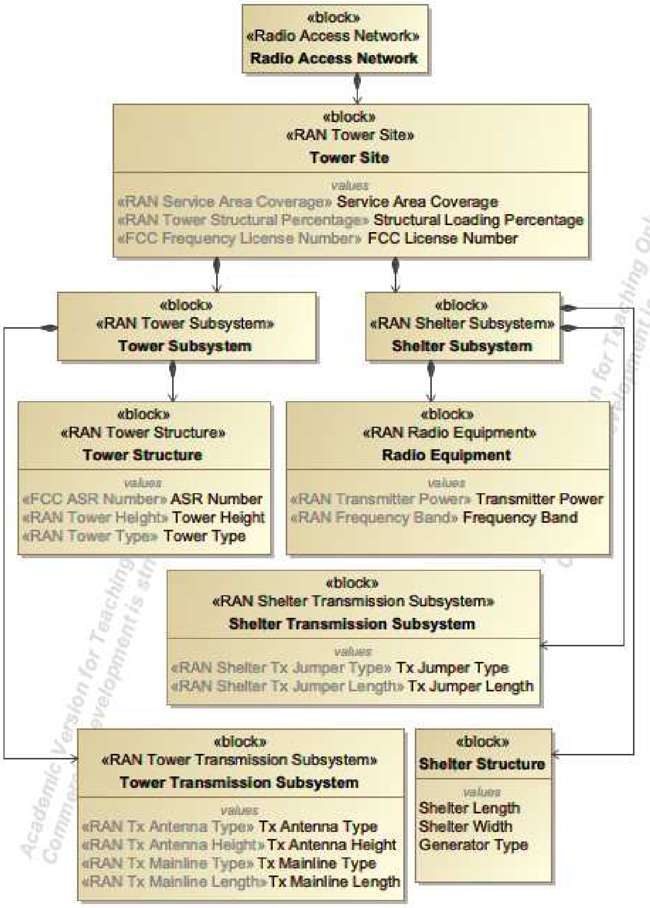

Fig. 7. Radio Access Network SysML Block Definition Diagram

Note that the "Shelter Structure" block and its value properties are not stereotyped. One key benefit of ontologies are their extensibility; they need not be complete in order to be useful. In fact, in many cases they are never complete as they are meant to describe domains of knowledge that are constantly expanding. In this example, certain terms related to the shelter were not captured in the developed ontology. However, instead of expanding the ontology, the team decided to forego tagging of the elements as they were not relevant to any cross-domain analysis planned. They are still part of the architecture, and should the need arise, they can still be tagged with ontologically relevant stereotypes, but the development team can invest in that expansion of the ontology when it is needed and does not need to describe everything in ontologies before any value can be extracted from the ontologies and the system aligned with them.

V. Discussion

Limitations of this research come from its implementation as a conceptual model. Use of NLP to provides matching will vary considerably depending on the approach used to determine similarity. This could limit the effectiveness of the matching as well as the amount of information provided as part of the expert verification step. For example, there may be ways to provide scoring of how "close" an ontology class is to a SysML model term, and there may be an opportunity to provide the expert with the top 3 options instead of a single match. These details must be addressed in future research. However, the authors argue that different similarity metric approaches may perform better or worse depending on the domains being used and the modeling culture of the teams using the tool. Therefore, the black box approach seems best at this point to allow for experimentation and flexibility to fit individual context.

The mentioned black box approach also enables flexibility as different methods and even technologies could be implemented in the process due to its modularity. If new and potentially better algorithms were to be found, they could be used instead of the ones presented in this paper, which provided future applicability of the presented concepts.

While the lack of full artificial intelligence automating the entire matching process may be seen as a limitation, it is important to restate that augmented intelligence that accelerates performance has considerable value. The authors see this approach as a way to bridge the gap between theoretically possible applications using ontologies that do not easily scale and large-scale design and modeling efforts. To achieve this, full automation is not required; it is only required that a method be developed to accelerate the mapping process. In fact, expert verification can be used to further train the NLP models to perform better over time, and in the process experts will also learn to trust the matching process more, so a slower approach to the use of artificial intelligence as an aid may pay dividends in trust in the future.

Furthermore, repeated and captured application of such processes will allow for the detection of patterns. Such patterns can then in turn be used through analysis to improve the functions underlying the presented approach. As such, machine learning technologies could also be considered moving forward. We argue that overall, while the conceptual nature of the presented work introduces and poses limitations, the removal and resolution of these constraints can be easily and flexibly addressed, which also directly connects the presented work and content to the future work and potential in the next section.

VI. Future Work and Conclusion

As mentioned above, while the work at hand is theoretical as of the time of this writing, there are different directions that can be pursued.

A. Future Work

Future work can actualize the conceptual framework presented in this paper. All portions of this method except for the expert verification step have the capacity to be automated. Where the method is able to automatically pull term lists and run comparisons using an NLP similarity module to present for expert verification, it has the potential to accelerate the mapping process considerably.

In addition, further research can be done on determining the optimal approach to achieve quality NLP results for matching. Ontologies can contain more than just a taxonomy of classes; they can also include many different axioms and relationships between classes beyond the taxonomical is_a relationship. This richer understanding of the classes could provide considerable performance improvement in the matching step between a requirement or SysML model term and an ontological term. For example, if an ontology specifies the different parts a shelter subsystem has and a SysML model lists those parts in a composite relationship with an oddly named parent block, it may be that the understanding of parthood from the ontology could give the matching algorithm enough confidence to present the parent block in the SysML model as a shelter subsystem. Further research into this approach as well as the many different NLP similarity and text matching approaches in this context is warranted.

Finally, given the instance related data that will be included in many requirement statements (such as 9 dB gain in the example use case given), additional work mapping that instance data to the ontology instead of just the class information would be worthwhile. Research using the DEFII framework to perform verification tasks has been explored [3], and the ability to capture key constraint data contained in the requirement text could provide additional model based verification functionality in the future.

B. Conclusion

This paper blends two areas of research – Digital Engineering knowledge representation through the use of ontologies and Natural Language Processing, both in relation to requirements engineering and in similarity metrics. This blend of research enables an augmented intelligence approach to accelerate the mapping of different digital engineering artifacts, namely textual requirements and SysML system models, to ontologies. Such an alignment allows for an ontological understanding of the system, which has positive implications for interoperability between different disciplines and a more holistic view of a system under design to improve and streamline development processes.

Acknowledgement

This research was sponsored by the Systems Engineering Research Center (SERC), a University Affiliated Research Center (UARC) housed at Stevens Institute of Technology.

References

[1] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowl. Acquis.*, vol. 5, no. 2, pp. 199–220, Jun. 1993, doi: 10.1006/knac.1993.1008.

[2] C. Hennig, A. Viehl, B. Kämpgen, and H. Eisenmann, "Ontology-Based Design of Space Systems," in *The Semantic Web – ISWC 2016*, vol. 9982, P. Groth, E. Simperl, A. Gray, M. Sabou, M. Krötzsch, F. Lecue, F. Flöck, and Y. Gil, Eds. Cham: Springer International Publishing, 2016, pp. 308–324. doi: 10.1007/978-3-319-46547-0_29.

[3] D. Dunbar, T. Hagedorn, M. Blackburn, and D. Verma, "Use of Semantic Web Technologies to Enable System Level Verification in Multi-Disciplinary Models," in *Advances in Transdisciplinary Engineering*, B. R. Moser, P. Koomsap, and J. Stjepandić, Eds. IOS Press, 2022. doi: 10.3233/ATDE220632.

[4] D. Dunbar *et al.*, "Driving digital engineering integration and interoperability through semantic integration of models with ontologies," *Syst. Eng.*, p. sys.21662, Mar. 2023, doi: 10.1002/sys.21662.

[5] D. A. Wagner, M. B. Bennett, R. Karban, N. Rouquette, S. Jenkins, and M. Ingham, "An ontology for State Analysis: Formalizing the mapping to SysML," in *2012 IEEE Aerospace Conference*, Big Sky, MT, Mar. 2012, pp. 1–16. doi: 10.1109/AERO.2012.6187335.

[6] J. S. Jenkins and N. F. Rouquette, "Semantically-Rigorous Systems Engineering Modeling Using SysML and OWL," p. 8, 2012.

[7] E. D. Liddy, "Natural Language Processing," in *Encyclopedia of Library and Information Science*, 2nd ed., NY: Marcel Decker, Inc., 2001.

[8] G. G. Chowdhury, "Natural language processing," *Annu. Rev. Inf. Sci. Technol.*, vol. 37, no. 1, pp. 51–89, 2003, doi: 10.1002/aris.1440370103.

[9] I. Beysolow and others, "What Is Natural Language Processing?," in *Applied Natural Language Processing with Python*, Springer, 2018, pp. 1–12.

[10] C. B. Achour, "Linguistic instruments for the integration of scenarios in requirements engineering," presented at the Proceedings of the Third International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'97), Barcelona, Catalonia, 1997.

[11] L. Zhao *et al.*, "Natural Language Processing for Requirements Engineering: A Systematic Mapping Study," *ACM Comput Surv*, vol. 54, no. 3, p. Article 55, 2021, doi: 10.1145/3444689.

[12] M. Vierlboeck, R. Nilchiani, and M. Blackburn, "Natural Language Processing to Assess Structure and Complexity of System Requirements," 2022, doi: 10.36227/techrxiv.20818984.v1.

[13] Z. S. Harris, "Distributional Structure," *WORD*, vol. 10, no. 2–3, pp. 146–162, Aug. 1954, doi: 10.1080/00437956.1954.11659520.

[14] B. B. Rieger, *On Distributed Representations in Word Semantics*. International Computer Science Institute Berkeley, CA, 1991.

[15] G. Nenadic, I. Spasić, and S. Ananiadou, "Automatic discovery of term similarities using pattern mining," presented at the CompuTerm 2002, 2002.

[16] L. Kof, "Natural Language Processing: Mature Enough for Requirements Documents Analysis?," Berlin, Heidelberg, 2005, pp. 91–102.

[17] C. Lipizzi, D. Borrelli, and F. de O. Capela, "A computational model implementing subjectivity with the'Room Theory'. The case of detecting Emotion from Text," *ArXiv Prepr. ArXiv200506059*, 2020.

[18] M. Vierlboeck, C. Lipizzi, and R. Nilchiani, "Natural Language in Requirements Engineering for Structure Inference--An Integrative Review," *ArXiv Prepr. ArXiv220205065*, 2022.

[19] D. Dunbar, M. Blackburn, T. Hagedorn, and D. Verma, "Application of the Digital Thread through the Digital Engineering Framework for Integration and Interoperablity," presented at the NDIA 25th Annual Systems and Mission Engineering Conference, Orlando, FL, USA, Nov. 2022.