Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo and M. Loper, eds.

TOWARDS SEMI-AUTOMATIC MODEL SPECIFICATION

David Shuttleworth

Modeling, Simulation and Visualization Eng. Dept.
Old Dominion University
5115 Hampton Boulevard
Norfolk, VA 23529

Jose J. Padilla

Virginia Modeling, Analysis and Simulation Center
Old Dominion University
1030 University Blvd
Suffolk, VA 23435, USA

ABSTRACT

This paper presents a natural language understanding (NLU) approach to transition a description of a phenomenon towards a simulation specification. As multidisciplinary endeavors using simulations increase, the need for teams to better communicate and make non-modelers active participants on the process increases. We focus on semi-automating the model conceptualization process towards the creation of a specification as it is one of the most challenging steps in collaborations. The approach relies on NLU processing of narratives, create a model that captures concepts and relationships, and finally provide a specification of a simulation implementation. An initial definition set and grammatical rules are proposed to formalize this process. These are followed by a Design of Experiments (DoE) to test the NLU model accuracy and a test case that generates Agent-Based Model (ABM) conceptualizations and specifications. We provide a discussion on the advantages and limitations of using NLUs for model conceptualization and specification processes.

1 INTRODUCTION

"Anyone who ventures a projection or imagines how a social dynamic . . . would unfold is running some model," (Epstein 2008). Epstein here suggests that everyone is capable of not only constructing a mental model but also simulating it (mentally) for prediction or explanation purposes. However, creating a computer-based model is a complicated task. In practice, problem stakeholders or domain experts (non-modelers) rely on rich descriptions or concept maps to capture concepts and their connections. The challenge is how to transition those mental maps to computer models. We need to both facilitate the learning process of modeling processes and facilitate the collaboration and engagement between domain experts and modelers.

There are several stages and approaches to developing models and transitioning them to simulations. For example, reference models are proposed as an initial stage capturing representations of a referent while also documenting assumptions and constraints (Tolk et al. 2013). At a specification stage of model development, modeling practitioners borrow the Unified Modeling Language (UML), for instance, to capture and communicate a conceptual model (Guizzardi and Wagner 2012; Reinhartz-Berger 2005). Both approaches, we argue, fall under engineering solutions not designed to support the elicitation, expression, and communication of models among multi-discipline teams. Furthermore, they are difficult to develop (Thalheim 2009).

Additionally, model development, and more importantly model communication, varies across domains and team structures. Describing models in the engineering or natural science disciplines/teams are typically based on observable systems/phenomena and identifiable/measurable variables grounded in quantitative observations or existing models. Describing a problem situation in an interdisciplinary effort requires more space for nuance, with domain experts in these activities likely unfamiliar with engineering-based

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

languages and tools or engineers unfamiliar with the approaches used by domain experts on the team. Communicating and characterizing a problem is indeed difficult in multi-domain efforts as participants describe their personal view of some phenomenon. Furthermore, moving from a problem description to a simulation introduces many transitions where context is lost and semantics degrade due to ambiguous interpretations among domain and modeling experts. Lastly, efforts to show how simulation models can represent phenomena often get lost in technical details frustrating and limiting domain expert participation.

The state of the art in computer modeling focuses on various stages of model development; often concentrating on the simulation-creation stages. The goal of this research is to start a discussion on how we can bridge the divide between domain experts and modelers in the process of creating simulation models. The approach proposes the elicitation of narratives and discussions in natural language between domain experts and modelers and the usage of natural language processing to identify components of a model. The team members will then transition to a model specification forgoing engineering-specific languages and tools.

2 BACKGROUND

Model development is an iterative process that often takes place between non-modelers (people interested in a modeling solution) and modelers (people providing a modeling solution). In this process, competing worldviews, knowledge level, and means of expression lead to ambiguous interpretations and frustrations among participants resulting in breaks in communication and engagement. Domain experts have diverse forms of expression characterized by variations in semantic depth and worldviews that compete with the modeling specialist's interpretation of these expressions and methods. It is the modeling specialist that is required to interpret the expert's expressed perception in order to facilitate model development. Unfortunately, this introduces the potential for ambiguities and inconsistencies that must be resolved in order to ensure models fulfill their objectives and the team is not decimated. We posit that natural language processing-based tools can bridge the gap by allowing non-modelers to create model conceptualizations and specifications. NLU/NLP models will make them active participants in a stage of the computer modeling process where they currently do not have the specialized training and expertise.

Natural Language Processing (NLP) primarily deals with how patterns in a corpora "reveal the syntactic structure of language" (Manning and Schutze 1999). A set of algorithms that make up NLP capabilities convert unstructured text to structured text. NLP uses machine learning (ML) to perform a variety of functions such as parsing sentences into individual parts of speech, examining the root form of words, and identifying named entities. NLU, a subset of NLP, refers to "semantic processing" resulting in a "deeper representation" (Barriere 2016). This is in contrast to traditional NLP approaches as NLU enables the identification of relationships among concepts and capture meaning. Recent approaches have invested in neural networks and deep learning as a shift away from traditional statistical approaches that focus on modeling of word sequences in a language to learning the distribution of words and determining sequence probability (Torfi et al. 2020).

Figure 1 illustrates the transitions from a description of a phenomenon to simulation specification. A narrative captures a description of the phenomenon. An NLU model converts the narrative to a collection concepts and relationships, which captures a model conceptualization. A model specification is then formed from this conceptualization, or from the narrative itself.

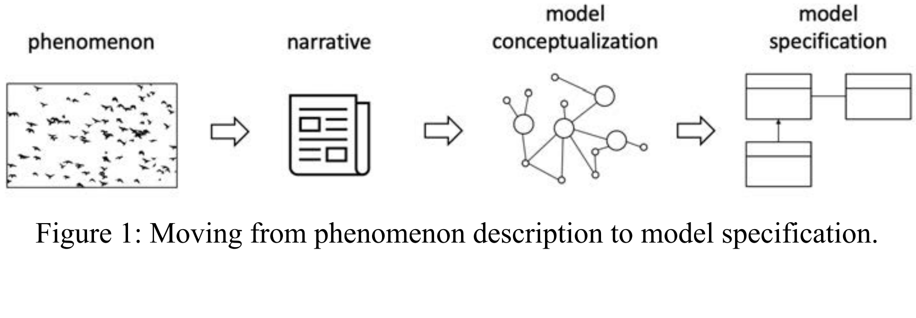

Figure 1: Moving from phenomenon description to model specification.

3 DEFINITIONS

To transition from narrative to conceptualization and specification, we need to define these terms. Let us consider that *concepts* and *relationships* are the two narrative elements required to create a model conceptualization. A concept is either an actor, a factor, or a mechanism that is relevant to the phenomenon of interest. Actors represent entities performing a function or are otherwise impacted by another object or entity. Factors are variables, and mechanisms represent action or an effect. Lastly, relationships are directed associations between concepts. A set of definitions is provided to guide the NLU annotation and model development process, but more importantly to explain the transition from descriptive narratives to model specification. It is important to note that the conceptualization and specification provided in this paper is focused on ABM. However, the process is extensible to other paradigms.

Definition 1 Conceptualization (of a model) represents a collection of concepts and relationships. Concepts are categorized into actors, factors, or mechanisms. The following partial function, adapted from Diallo (2010), formally represents the *concepts* of a conceptualization.

$$F(S) = \begin{cases} \mathit{\Omega} \text{ if } S \text{ is an actor} \\ \Pi \text{ if } S \text{ is a factor} \\ M \text{ if } S \text{ is a mechanism} \end{cases} \tag{1}$$

Individual concepts are defined as $\omega \in \mathit{\Omega}$, $\pi \in \Pi$, and $\mu \in M$, which states that each concept is an element within a set of actors, factors, or mechanisms.

Definition 2 If $s_i, s_j \in S$, then $G(s_i, s_j)$ is a *directed relationship* between two concepts. The directed relationship is a non-commutative binary relation; in other words, $G(s_1, s_2) \neq G(s_2, s_1)$. These nine constraints require that an actor, factor, or mechanism modify another actor, factor, or a mechanism. A mechanism may either be affected by an actor or have an effect on an actor, but a mechanism cannot be executed by an factor. Lastly, an actor may have an influence over another actor or have some other association or dependency.

$$\begin{matrix} \mathrm{G}(\omega, \pi) & \mathrm{G}(\omega, \mu) & G(\omega_i, \omega_j) \\ \mathrm{G}(\pi, \omega) & \mathrm{G}(\pi, \mu) & G(\pi_i, \pi_j) \\ \mathrm{G}(\mu, \omega) & \mathrm{G}(\mu, \pi) & G(\mu_i, \mu_j) \end{matrix} \tag{2}$$

For example, a sheep (*actor*) will consume (*mechanism*) grass (*actor*) resulting in an increase in energy (*factor*). The directional relationship is from sheep to consume and from consume to grass. Sheep is described by the *factor*, energy.

Definition 3 A narrative contains nouns, verbs, and modifiers. ℕ represents the set of nouns; 𝕍 is the set of verbs; and 𝔸 is the set of modifiers, which are either adjectives and adverbs. Therefore, the set of these narrative elements are represented by

$$\begin{matrix} n \in \mathbb{N} \\ v \in \mathbb{V}. \\ a \in \mathbb{A} \end{matrix} \tag{3}$$

Nouns that refer to entities considered performers and places are annotated as actors. Factors are quantitative or qualitative variables that describe, modify, or provide a value associated to an actor or mechanism. Mechanisms are identified provided an action verb associated to an actor as a subject or object. Relationships are annotated if there is an explicit or easily inferred dependency.

Definition 4 Specification (agent-based model), in the form of agents, attributes, and rules, is the implementation of actors, factors, and mechanisms. Agents, attributes, and rules follow the same syntactical rules described previously. Agents can be resolved from nouns, and attributes are either adjectives or adverbs as they describe a variable or characteristic of an agent. The following equations add additional constraints to the transition from model conceptualization to ABM components: agent, attribute, or rule. The intent here is to not remove concepts that fail the constraints but to facilitate completing the model specification.

$$\omega_i \text{ is an agent if } \exists\{n_i | G(n_i, v_j) \vee G(n_i, a_k)\} \tag{4}$$

$$\pi_i \text{ is an attribute if } \exists\{a_i | G(n_j, a_i) \vee G(v_k, a_i)\} \tag{5}$$

$$\mu_i \text{ is an rule if } \exists\{v_i | G(n_j, v_i) \vee G(v_i, n_j) \vee G(v_i, a_k)\} \tag{6}$$

While this body of research is focused on ABM, Definition 4 can be generalized and applied to other modeling paradigms. Table 1 provides a non-exhaustive set of examples of how actors, factors, and mechanisms would be implemented in the ABM, Discrete Event Simulation (DES), and System Dynamics (SD) paradigms.

Table 1: Relating concepts to paradigm-specific components.

Concept	ABM	DES	SD
Actor	Agent	Entity	Population
Factor	Attribute	Rate	Number
Mechanism	Rule	Process	Causality

4 CLASSIFYING CONCEPTS FROM A NARRATIVE

To understand how to train NLU models, and generate a conceptualization and a specification of a phenomenon via a narrative, we look into context-free and dependency-based grammatical structures as means to parse sentences. Chomsky (1956) proposed constituency grammar, or context-free grammar and phrase structures for parsing sentences. Constituency grammar simplifies a sentence into phrases such as noun phrases (NP), verb phrases (VP), adjective phrases (AdjP), and adverb phrases (AdvP) as shown in Figure 2.

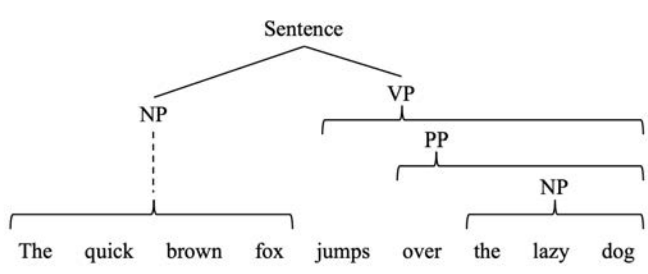

Figure 2: Example of a constituency grammar.

While efficient at providing a finite-state grammar, more nuance is needed for identifying and relating individual concepts. Hays (1964) proposes a dependency theory and explains impact of correspondence on ambiguity. His classical example phrase "they are flying planes" has two meanings. Either *they* correspond to *flying planes*, or *they* refers to a pilot *flying* a *plane*. The key determinant on deciding which connotation is relevant is whether *are flying* is a VP or *flying planes* is a NP.

Dependency grammar also focuses on the relationships among terms (Nivre 2010). A directed graph can be formed such that an edge from node *i* to node *j* denotes $word_j$ depends on $word_i$ (Debusmann and Kuhlmann 2010), and dependency grammar's reliance on directed associations make it well suited for customizing an ML model where directed relationships among concepts are desired. Constituency grammar offers a more layered approach and is not restrained in a one-to-one mapping, but lacks the detail and nuance needed. Dependency grammar is more efficient in developing structures. (Osborne 2019). An example of sentence parsing based on dependency grammar is shown in Figure 3.

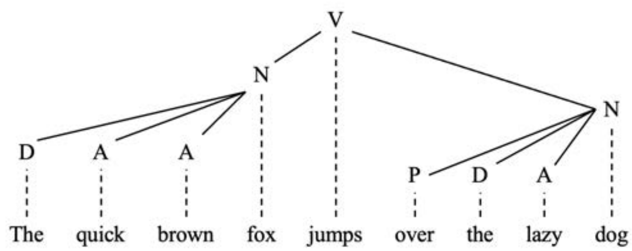

Figure 3: Example of a dependency grammar.

Natural language is complex, inconsistent, and ambiguous (Chowdhary 2020). While one may be able to extract concepts and relationships solely on a prescriptive set of grammatical rules, ML models are able to do this automatically provided a model is sufficiently trained. ML, more specifically, NLU can facilitate the creation of a candidate knowledge graph directly from a narrative while maintaining a semantic commitment to the narrative.

IBM's Watson Knowledge Studio (WKS) is used in this body of research for training the NLU model in an iterative, semi-supervised approach. Even though WKS is proprietary and considered a black box, it was chosen given the simple user interface requiring little programming skills and rapid experiment execution. Figure 4 illustrates the pipeline for developing an NLU model.

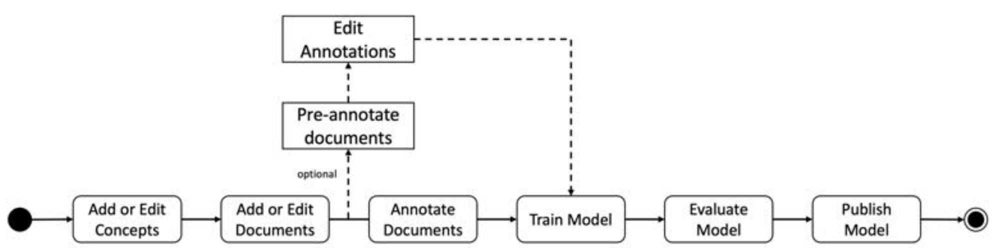

Figure 4: Corpus collection and NLU model development process.

The first step in utilizing WKS is to define the types of concepts and relationships that one wishes to extract from a narrative. An initial corpus is then annotated manually so that an NLU model can be trained provided the annotated corpus and evaluated against an annotated test corpus. Once a model is trained, it may be used to pre-annotate additional documents reducing manual annotation. Upon using a bootstrapping approach, the results are manually inspected for accuracy and modified as needed following the dependency grammar described previously. Additional narratives and subsequent annotations are added to the corpus incrementally until a desired accuracy is achieved. The model can be deployed so that it can be accessed and exercised outside the WKS environment through an application interface coded in Python, JAVA, etc.

5 MODEL CONCEPTUALIZATION

A model conceptualization, extracted from a narrative using an NLU model, is shown in Figure 5. It provides the computer annotated actors, factors, and mechanisms from a journal article excerpt on group problem solving (Carletti et al. 2020). Actors and factors are identified in addition to several types of relationships: *hasFactor*, *partOf*, and *attenuatedBy*. *Capabilities* is characterized by the adjective, *problem-solving*, and *problem-solving* is qualitatively assigned and attenuated by the adverb, *better*. These factors describe actors in a qualitative sense, but factors can also include quantitative or empirical data. Other relationships include associations between actors and mechanisms and factors and mechanisms.

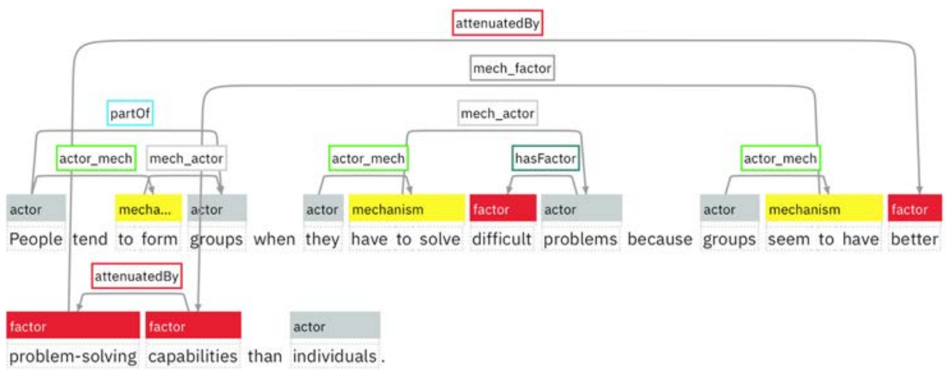

Figure 5: Example actor, factor, and relationship annotations of a sentence.

It is important to note that this conceptualization identifies actors, and corresponding factors and mechanisms, conceptualization and is meant to be independent of a modeling paradigm. The identification of concepts and relationships among concepts is the primary objective of this step. The way the relationships are expressed is important. For instance, from Figure 5 we can extract relationships between factors/mechanisms of the population. Group formation as a function of problem difficulty is one example of such relationship. In this case, we can infer that the more difficult the problem is the higher the likelihood groups are formed. In other cases, the identification of resources and waiting times would indicate a discrete event model. The identification of actor aggregation and/or high incidence of actors performing mechanisms may suggest an ABM would be most appropriate. It is important to reiterate that a model conceptualization is paradigm agnostic, and that in reality one or more paradigms may be utilized to transition a model conceptualization to model specification. In this body of research, we trained a NLU model to generate an ABM specification by recognizing agents, attributes, and rules.

6 MODEL SPECIFICATION (AGENT-BASED MODEL)

This approach extends the approach described by Padilla et al. (2019) where agents, attributes, and rules were extracted from a narrative, but the research did not address relationships between concepts. Further refinement is also achieved by following a more strict grammatical ruleset that dictates a more consistent corpus annotations for semi-supervised training.

An agent is an individual unit or an aggregation of units. An attribute may describe a variable characteristic of an agent; it may provide an additional layer of characterization to another attribute; or it may describe a variable associated to a rule. Rules in ABM describe agent behaviors and interactions with other agents or the surrounding environment. For example, a person may be specified by their age and have an affinity for other persons specified by the rule, congregation. A rule regarding reaction of a prey to a close predator may be specified by the speed of reaction.

Table 2 lists several examples of each ABM component. These are simple descriptions of agents, attributes, and rules. By identifying and related ABM components to each other, we can arrive at a richer model specification.

Table 2: Characterization of an ABM.

ABM Component	Examples
Agent	Person
	Animal
	Virus
	Environment
Attribute	Age
	Speed
	Size
Rule	Avoidance
	Cohesion
	Reaction

A model specification can be represented in several ways. Upon processing a narrative and developing a model conceptualization, the concepts are transitioned to the appropriate paradigm. In this case, they become agents, attributes, and rules, these elements may be visualized as a graph of nodes and edges not unlike an ontology or concept map. This enables a visual check in an easy to read format and uncover relationship patterns that may be useful in accepting or rejecting the design. These elements can easily be translated to a model specification as shown in Figure 6. The example shown here is the result of a manual process, but it is expected this translation can be achieved via an interchange format such as JavaScript Object Notation (JSON) or eXtensible Markup Language (XML). It is important to note that the specification here is derived from the same narrative that the produced the conceptualization. However, the specification can be generated from the subset of concepts and relationships that have the most information available.

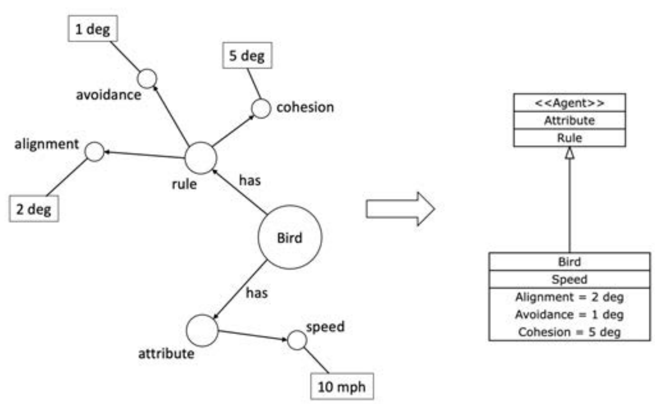

Figure 6: Transitioning from knowledge graph to UML class diagram.

7 NLU IN SUPORT OF GENERATING A MODEL SPECIFICATION

Two Design of Experiments (DoE) were executed to assess how well the NLU model performed in generating ABM conceptualizations and specifications. The metrics used for evaluation are precision, recall, and a combination of precision and recall. Precision is the ratio of true positives and the total of true positives and false positives. Recall is the ratio between true positives and the total of true positives and false negatives. F1 is a function of both precision and recall given by

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \tag{7}$$

Each DoE contained five levels of training narratives ranging from 10 to 30 in increments of five. The training corpus was composed of excerpts from journal articles from the Journal of Artificial Societies and Social Simulation (JASSS 2021) containing descriptions of ABMs. The first DoE, DoE A, testing corpus was also from JASSS, whereas the second DoE, DoE B, testing corpus contained descriptions of case files associated to insider threat activity. The DoE B was executed in order to evaluate how well the models would perform when tested against a corpus from an unrelated domain. The run matrix and associated test results from the DoE A is provided in Table 3.

7.1 DoE to Evaluate Specification Description Performance

For DoE A, the scores' rate of increase diminishes after 20 documents are used for training. At this level, the model is correctly identifying *agents* and *attributes*, and satisfactorily identifying *rules*. The *relationship* scores demonstrate a similar trend, but illustrate poorer performance with the exception of the *relationship* score between an *agent* and an *attribute*.

Table 3: Results from DoE A.

Exp ID	Documents		Agents			Attributes			Rules		
	Train	Test	F1	Prec.	Recall	F1	Prec.	Recal	F1	Prec.	Recall
1	10	5	0.59	0.71	0.51	0.48	0.56	0.41	0.35	0.44	0.29
2	15	5	0.67	0.76	0.59	0.51	0.53	0.49	0.41	0.49	0.36
3	20	5	0.70	0.83	0.60	0.60	0.65	0.56	0.50	0.59	0.43
4	25	5	0.69	0.78	0.61	0.55	0.55	0.54	0.51	0.53	0.49
5	30	5	0.70	0.73	0.67	0.57	0.57	0.57	0.49	0.52	0.47

Exp ID	Documents		Agent-Attribute			Agent-Rule			Rule-Agent		
	Train	Test	F1	Prec.	Recall	F1	Prec.	Recal	F1	Prec.	Recall
1	10	5	0.32	0.43	0.25	0.20	0.32	0.15	0.13	0.22	0.09
2	15	5	0.38	0.44	0.34	0.30	0.34	0.27	0.26	0.47	0.18
3	20	5	0.54	0.64	0.47	0.38	0.45	0.33	0.32	0.56	0.23
4	25	5	0.45	0.47	0.44	0.36	0.36	0.37	0.33	0.45	0.26
5	30	5	0.47	0.45	0.49	0.36	0.35	0.37	0.33	0.45	0.26

An example of a poor result from the fifth experiment is shown in Figure 7. There are several absent and incorrect annotations in this example. *Inner* should have been classified as an attribute while *ring* should be an agent. This illustrates the ambiguity associated with natural language as *ring* can be either a noun or verb depending on the context. *They* and *want to sell* are not classified in the second half of the sentence, which is indicative of an increase in misclassifications as noted by the lower precision and recall scores.

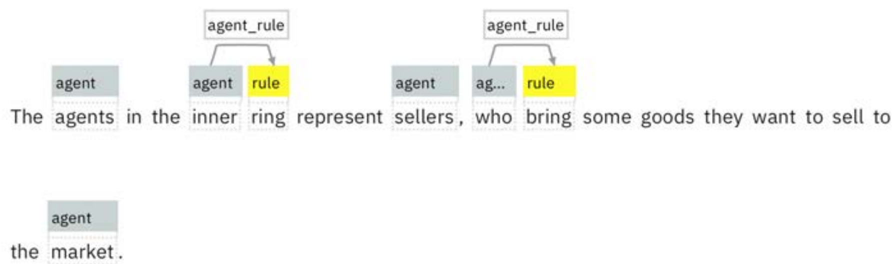

Figure 7: Example result from DoE A, experiment five.

In comparison, the same sentence from the third experiment, which had slightly better scores, is shown in Figure 8. This experiment provides a much different result; although it is still not ideal. Note that there are misclassifications associated with *represent* and *market*, but the model did classify *want to sell*. It is interesting that in the previous example, the *who-bring* relationship is correctly classified, while here, the *bring-goods* relationship is identified. The model does result in some ambiguous results, but it does perform well in identifying the concepts, *agents*, *attributes*, and *rules*. It also further validates the feasibility of the approach. More importantly, it enables us to make adjustments to the overall approach in subsequent studies.

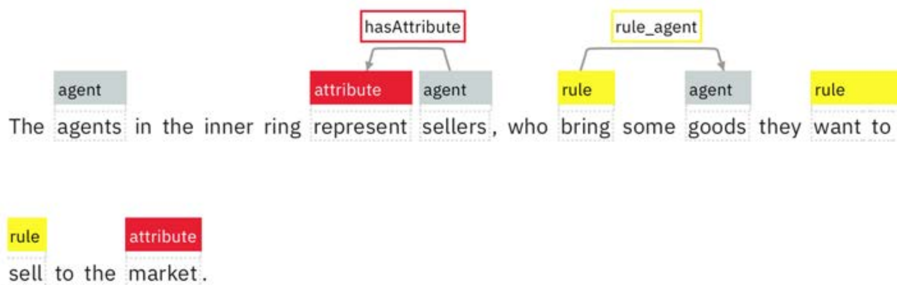

Figure 8: Example result from DoE A, experiment three.

7.2 Model Specification Derived from a Non-ABM Description

The second DoE was conducted using the same training sets but tested against five documents from an unrelated domain, insider threat case files. The results table is omitted here for succinctness, but DoE B resulted in poorer performance, which is an indication that domain-dependent NLU models may be required. Secondly, since the five test documents were annotated after the first experiment was completed, annotation bias may have led to poorer test performance.

The NLU model was also used to extract concepts and relationships from an narrative describing the refugee crisis on the Island of Lesvos (Jauhiainen and Vorobeva 2020). The following is an excerpt used to illustrate the ability of the NLU model to identify ABM specification elements:

"To make the dispersal of the illness slower, the restaurants, hotels and other accommodation were closed, the camps were temporarily closed for external persons and the full lockdown of the country was exercised."

The excerpt details a problem situation containing several elements: political pressure, asylum seekers, anti-migrant beliefs, violence, events, and disease. In this short example, there are many interrelated concepts that would be challenging to capture initially in a visual format without first expressing in natural language. This excerpt was submitted as a new document and the pre-annotator was applied to visualize the results, which is shown in Figure 9. There are several complex verb phrases that were correctly classified, such as *were closed* and *was exercised*, but the model misclassified *slower* as an agent. Additionally *hotels* should have a relationship with *were closed*.

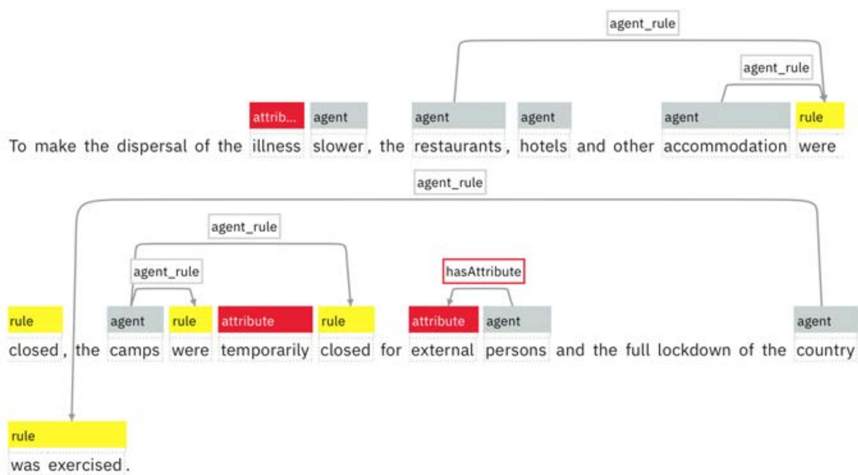

Figure 9: Example of agents, attributes, and rules from the refugee use case.

The output from the model can be visualized as a model conceptualization and converted to a model specification, which is shown in Figure 10.

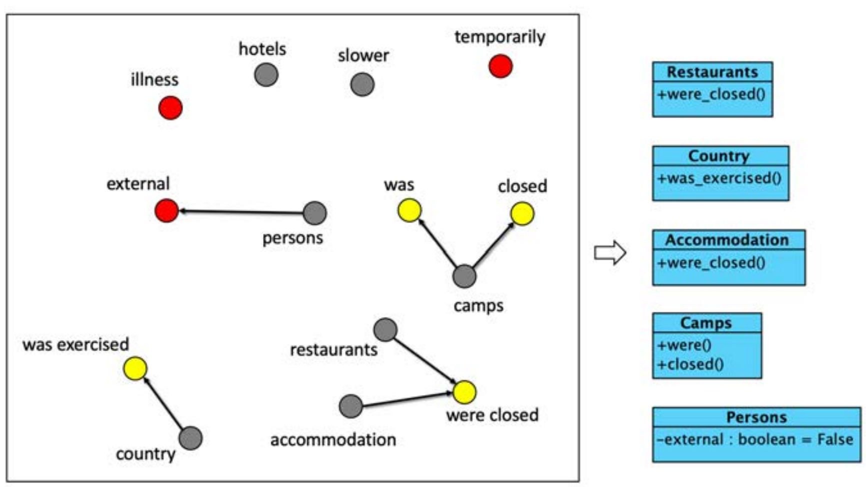

Figure 10: Transition from Model Conceptualization to ABM specification.

In this case, only the agents with relationships to rules and attributes are shown in the class diagram. Agent attributes are shown as class attributes, and agent rules are shown as operations or methods. The orphaned agents, attributes, and rules are shown in the graph and may be used as to identify where additional information is required. For example, it is likely that *illness* should be related to *dispersal* (not classified) and *slower* to quantify an infection rate.

8 DISCUSSION AND FUTURE WORK

The results highlight that NLU models provide a means of facilitating modeling to non-modelers by semi-automatically constructing a model conceptualizations and specifications from descriptions of a phenomenon. In this research, two separate NLU models were used to generate conceptualization and specifications of ABMs. Agents, attributes, and rules are derived from actors, factors, and mechanisms, respectively. Other modeling paradigms subscribe to different concept types such as resources, processes, populations, and causal loops. Furthermore, having a set of alternative NLU models spanning multiple knowledge domains and modeling paradigms could provide a modeling decision layer where each NLU model is applied to a target narrative.

It may be desirable to have a narrative or conversational template to capture the minimal viable set of elements required from a phenomenon description. For example, if a narrative contains orphaned actors, factors, or mechanisms, a modeler (human or computer-based) would conclude that either the orphaned concept is superfluous or additional information is missing. If there is no indication of a factor having a relationship to an actor, then the modeling assistant should elicit additional detail from the subject matter expert or existing data repositories. A digital modeling assistant, for instance, would also elicit quantitative and qualitative data if data that should be associated to extracted factors, or attributes, is missing. This process could be completed directly by the modeler visually by manipulating a concept map, or done assisted through more advanced machine reasoning techniques utilizing an ontology.

The specification up to this point relies on an object-oriented view of ABM. It is important to consider model formalisms such as the Discrete EVent System (DEVS) specification. Formalisms may facilitate the training process, for instance. Mittal et al. (2009) demonstrated populating a formal DEVS specification from a combination of restricted NLP, requirement specifications, and modeling language. This set of specifications were translated into XML to enable simulation integration along with other similarly developed implementations. In this research, the collection of concepts and relationships are easily exported in JSON format, which can be translated to formal specifications such as DEVS either directly from JSON or via UML as explored in existing literature (Mittal et al. 2009).

Per WKS standards, enough documents were included in the training, diminishing returns on performance metrics in DoE A leads us to believe that the annotations themselves need to be adjusted and/or we need better algorithms to obtain the desire outcome. With regard to DoE B, it is hypothesized that variations in writing style, annotation biases, and possible domain confusion led to poorer performance. While additional validation is required, there appeared to many more instances where two nouns such as *computer network* were initially annotated as two nouns, but the model decoded the two words as an attribute and agent. Additional categories of concepts may be required that take into account other types of modifiers and part-whole relationships. For example, in this case, *computer* is a type of *network*. However, this also establishes the need for techniques to uncover semantics and intent, which may require a modeler in the loop to address cases of ambiguous terms. Another shortfall is the lack of complete phenomena descriptions in journal articles. Authors tend to jump directly into how the model is implemented versus describing the problem, which indicates the lack of fully describing the conceptualization.

Lastly, WKS offers very little algorithm description as it applies to their implementation. Future implementations require the use of a more open platform so as to have more insight and control into the specific training methods used. The inclusion of standard dependency treebanks to support an automated annotation process used in conjunction with the grammatical rulesets to refine the relationships should make the process overall more efficient.

ACKNOWLEDGMENTS

This research is funded by grant number N000141912624 by the Office of Naval Research through the Minerva Research Initiative; none of the views reported in the study are those of the funding organization.

REFERENCES

Barriere, C. 2016. *Natural Language Understanding in a Semantic Web Context*. Cham: Springer.
Carletti, T., A. Guarino, A. Guazzini, and F. Stefanelli. 2020. “Problem Solving: When Groups Perform Better Than Teammates”. *Journal of Artificial Societies and Social Simulation* 23(3):1-16.
Chowdhary, K. R. 2020. Fundamentals of Artificial Intelligence. New Delhi: Springer.
Chomsky, N. 1956. "Three Models for the Description of Language". *IEEE Transaction on Information Theory* 2(3):113-124.
Debusmann R. and M. Kuhlmann. 2010. Dependency Grammar: Classification and Exploration. In *Resource-Adaptive Cognitive Processes. Cognitive Technologies*, edited by M. Crocker and J. Siekmann J., 365-388. Berlin: Springer.
Diallo, S. 2010. Towards a Formal Theory of Interoperability. PhD. thesis, Department of Computational Modeling and Simulation Engineering, Old Dominion University, Norfolk, Virginia.
Epstein, J. 2008. “Why Model”. *Journal of Artificial Societies and Social Simulation* 11(4):1-12.
Frerichs, L., T. L. Young, G. Dave, D. Stith, G. Corbie-Smith, and K. H. Lich. 2018. “Mind Maps and Network Analysis to Evaluate Conceptualization of Complex Issues: A Case Example Evaluating Systems Science Workshops for Childhood Obesity Prevention”. *Evaluation and Program Planning* 68(1):135-147.
Guizzardi, G., and G. Wagner. 2012. “Tutorial: Conceptual simulation modeling with Onto-UML”. In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 52-66. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
Hays, D. G. 1964. “Dependency Theory: A Formalism and Some Observations”. *Language* 40(4):511-525.
Jauhiainen, J. and E. Vorobeva. 2020. “Asylum Seekers and Migrants in Lesvos, Greece”. Technical Report No. 15, Department of Geography and Geology, University of Turku, Turku, Finland.
JASSS. 2011. “About JASSS”. http://jasss.soc.surrey.ac.uk/admin/about.html, accessed 15th June.
Manning, C. and H. Schütze. 1999. Foundations of Statistical Natural Language Processing. Cambridge: MIT Press.
Mittal, S., J. L. Risco-Martin, and B. P. Ziegler. 2009. “DEVS/SOA: A Cross-Platform Framework for Net-centric Modeling and Simulation in DEVS Unified Process”. *Simulation* 85(7):419-450.
Nivre, J. 2010. “Dependency Parsing”. *Language and Linguistics Compass* 4(3):138-152.
Osborne, T. 2019. *A Dependency Grammar of English: An Introduction and Beyond*. Amsterdam: John Benjamins Publishing Company.
Padilla, J. J., D. Shuttleworth, and K. O’Brien. 2019. “Agent-Based Model Characterization using Natural Language Processing”. In *Proceedings of the 2019 Winter Simulation Conference*, edited by N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, pp. 560-571. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
Reinhartz-Berger, I. 2005. “Conceptual Modeling of Structure and Behavior with UML – The Top Level Object-Oriented Framework (TLOOF) Approach”. In *Proceedings of the 2005 International Conference on Conceptual Modeling*, edited by L. Delcambre, C. Kop, H.C. Mayr, J. Mylopoulos, and O. Pastor, 1-15. Berlin: Springer.
Thalheim, B. 2010. “Towards a Theory of Conceptual Modelling”. *Journal of Universal Computer Science* 16(20):3102-3137.
Tolk, A., S. Diallo, J. Padilla, and H. Herencia-Zapana. 2013. “Reference Modeling in Support of M&S – Foundations and Applications”. *Journal of Simulation* 7(2):69-82.
Torfi A., R. A. Shirvani, Y. Keneshloo, N. Tavaf, and E. A. Fox. 2020. “Natural Language Processing Advancements by Deep Learning”. *ArXiv*. https://arxiv.org/pdf/2003.01200.pdf, accessed 9th April.

AUTHOR BIOGRAPHIES

DAVID SHUTTLEWORTH is on staff at the Johns Hopkins University, Applied Physics Lab. He is a Ph.D. candidate in M&S at Old Dominion University. His research interests include agent-based modeling, artificial intelligence, conceptual modeling, and operations analysis. His research area is in the use of Machine Learning to semi-automate model development from narratives. His email address is dshut002@odu.edu.

JOSE PADILLA is a Research Associate Professor at the Virginia Modeling, Analysis and Simulation Center at Old Dominion University. His research focuses 1) on the combination of methods and technological innovations to make simulations accessible to non-simulationists and 2) on the application of those methods to study areas like cybersecurity and forced migration. His email address is jpadilla@odu.edu.