.*Proceedings of the 2019 Winter Simulation Conference*
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

AGENT-BASED MODEL CHARACTERIZATION USING NATURAL LANGUAGE PROCESSING

Jose J Padilla

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

David Shuttleworth

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

Kevin O'Brien

Computer Science Department
Old Dominion University
5115 Hampton Boulevard
Norfolk, VA 23529, USA

ABSTRACT

This paper reports on Natural Language Processing (NLP) as a technique to analyze phenomena towards specifying agent-based models (ABM). The objective of the ABM NLP Analyzer is to facilitate non-simulationists to actively engage in the learning and collaborative designing of ABMs. The NLP model identifies candidate agents, candidate agent attributes, and candidate rules all of which non-simulationists can later evaluate for feasibility. IBM's Watson Natural Language Understanding (NLU) and Knowledge Studio were used in order to annotate, evaluate, extract agents, agent attributes, and agent rules from unstructured descriptions of phenomena. The software, and related agent-attribute-rule characterization, provides insight into a simple but useful means of conceptualizing and specifying baseline ABMs. Further, it emphasizes on how to approach the design of ABMs without the use of NLP by focusing on the identification of agent, attributes and rules.

1. INTRODUCTION

The characterization of a problem, the conceptualization of a model, and a specification of a simulation provide challenges ranging from knowledge domain and methodological familiarity to computational modeling abilities. Methodological differences bring one of the major challenges. For instance, while ethnographers study phenomena via rich and unstructured narratives, engineers rely on analysis towards establishing overarching systemic structures or underlying rules of behavior. As each approach brings its own worldview and embedded assumptions, the challenge becomes bridging methodological approaches but more importantly, empowering subject matter experts, or non-simulationists in general, to participate in the modeling process. Often, simulationists elicit information from subject-matter experts about a phenomenon of interest. Simulationists then build the models and socialize their results with non-simulationists. This approach leaves the non-simulationist out of the design process as the modeler's jargon and worldviews do not find their way to the subject matter expert. Further, this process leads to models not being use by subject matter experts, and in the worst case, it leads to their distrust of models. Padilla et al. (2018) highlights some of the challenges of bridging the methodological divide between ethnographers and simulationists towards the creation of models. This work suggests the conceptualization of a model as a

978-1-7281-3283-9/19/$31.00 ©2019 IEEE

starting point to arrive to a methodological consensus to understand and characterize the phenomenon of interest. However, conceptualization may start from unstructured narratives from which variables and rules must be identified.

Describing some phenomenon using natural language creates the opportunity for ambiguity and misinterpretations. Furthermore, each person observes the world in a unique way and applies their own biases to the problem conceptualization. These multiple perspectives with the added colorfulness of expressiveness introduce opportunities for misunderstanding among multiple stakeholder views. Yet, according to Tolk et al. (2013), expressions of a conceptualization may be a collection of sentences representing the modelers view of reality and/or the observed problem situation. This suggests the need to arrive to a collection of sentences that can be disambiguated to later be "translated" to a model, in this case to an agent-based model (ABM).

Unlike discrete-event simulation or system dynamics, where we can rely on simple diagrams capturing a sequence of steps or causal loops respectively, ABM conceptualization relies on *ad hoc* approaches that vary per modeler. Furthermore, the conceptual model of an agent-based models tend to be closely associated to implementation, such as the Unified Modeling Language (UML) or Systems Modeling Language (SysML), an extension of UML. Other efforts such as the development of the Overview, Design concepts and Details (ODD) specifications are not used for conceptualization but for capturing the implementation specification (simulation blueprints). The lack of an agreed to approach for agent conceptual modeling coupled with the specialized approaches that do exist are entry barriers into developing simulations.

Moreover, in some instances, ABM specifications are created after the model has been developed bypassing a conceptualization process. As such, recognizing the ABM conceptualization process is a challenge not only generates awareness towards developing new approaches that bridge communities but also facilitate the teaching and learning to the non-simulationists community in general. The proposed approach on this paper focuses on facilitating the specification of a model from a description of a phenomenon described in an unstructured narrative to familiarize non-simulationists with ABM design elements, mainly agents, agent attributes and rules.

The paper is organized thusly: Section 2 presents an overview on the subject of agent specification, Section 3 presents natural language processing, Section 4 describes the ABM NLP Analyzer, and Section 5 provides a discussion and a way forward.

2. ABM SPECIFICATION

There have been several efforts to introduce approaches to modelling complex, adaptive systems with an intent to simulate the phenomenon as an ABM. The obvious are the modeling languages, such as UML and SysML. These modeling languages are more closely aligned with an implementation of a model, therefore is related to the simulation model specification as opposed to a conceptual model artifact.

As UML is well suited for object-oriented modeling and simulation design, it lends itself well to specifying in detail what the agents are and how they interact with each other and with the environment. There are other similar examples in the literature; Notably, the Journal of Artificial Societies and Social Simulation (JASSS) contains several examples of conceptual model development. Bersini (2012) advocates for an object-oriented approach relying on class, sequence, state, and activity diagrams, which is consistent with SysML artifacts. Bersini advocates separating behavioral modeling from the agent object model. More precisely, the behaviors are aggregated and associated to an agent via a composition relationship. This allows them to be introduced in state diagrams. It may not be immediately obvious, but there are triggers in both the state and activity diagrams. This is an important aspect of modeling agents as it begins to define rule sets. There is an immediate shortcoming: the ability to adapt or model sets of states, which are fundamental aspects of ABM and complex systems. This may be addressed as a series of state and activity diagrams, but this is a work-around solution at best.

More recently, Siebers and Onggo (2014) proposed using UML for a graphical representation for ABM. Their paper focused on state diagrams and some on use case diagrams, which Bersini did not address. The notion of history, or agent memory, in the state machine diagrams is briefly discussed, but the rationale for including them is not thoroughly discussed. The history as well as the concept regarding model patterns introduces additional complexity to the model development process. These model patterns can indicate centralized, decentralized, or hierarchal designs.

Modelling Agent systems based on Institutional Analysis (MAIA) was developed as an extension of the Institutional Analysis and Development framework (IAD) as a framework for agent-based social simulation. MAIA has fives structures as described by Ghorbani, et al. (2013): collective, constitutional, physical, operational, and evaluative. The collective structure contains the agents, their attributes, and rules the agents follow based on criteria. The constitutional structure is the social context, and the physical structure is the physical characteristics of the system. The operational structure contains the system dynamics, and the evaluative structure consists of measures used in validation and evaluation. These structures may be represented, and associations described through a SysML package diagram containing classes representing the contents of each structure.

The ODD protocol was updated in 2010 (Grimm et al. 2010) to address a few aspects from the original format from 2006. The ODD was not designed for conceptualization, it was designed for capturing model descriptions. ODD was initially developed to support ABM of ecological systems, but has since been expanded to other domains within the social sciences (Grimm et al. 2017). Another extension, termed ODD-P includes provenance information, which includes how information was included in a model (Reinhardt et al. 2018). This information is useful in documenting the evolution of a model, therefore the knowledge of how assumptions were made or how models were reconciled is retained. The issue from a conceptualization point of view is that ODD is a process to document or catalog a model. While this is helpful in many aspects, it does not suggest a mechanism to conceptualize a problem or system.

Neither the modeling languages or the referenced processes support the elicitation of knowledge from the non-simulationist who should be able to describe a phenomenon in their own way, and they are more associated to an implementation, which requires a specialist well versed in agent-based modeling. A mechanism to reason over the description, which may be a narrative or even a series of text snippets resulting from a brainstorming session is needed so that the non-simulationist is an active member in the modeling process. Table 1 briefly lists the advantages and disadvantages of each. In summary, modeling languages when following the associated rigorous standards represent a steep learning curve, which is a barrier to the non-specialist. However, they are unambiguous, but are also closely associated to implementation and software development. The process frameworks such as ODD are easy to learn and overcome some of the entry barriers, but they are typically associated to specific domains such as ecology. The third technique, Natural Language, will be discussed in detail in the next section.

Table 1: Comparison of modeling techniques.

Technique	Example(s)	Advantages	Disadvantages
Modeling Languages	UML SysML BPMN	Formal, Unambiguous	Steep learning curve Typically associated to implementation and software development
Modeling Frameworks, Protocols	ODD and variants MAIA	Easy to use and learn	Has to be tailored for individual domains
Natural Language	Descriptive Text	Widely applicable, Flexible	Informal Ambiguous

3. NATURAL LANGUAGE PROCESSING (NLP)

Prior to NLP, the equivalent concept of natural language understanding was introduced over 55 years ago as a way for providing a computational method to comprehend human language (Martinez 2010). By taking an empiricist approach, it is asserted that the structure of language can be learned through a general model and applying statistical methods to machine learning techniques given an observed set of language examples. The alternative approach to NLP is a rationalist approach, where language cognition is not learned but encoded in one's genome. The statistical, or empiricist, approach postulates that there is some natural cognitive notion of language that is general purpose (Manning and Schutze 2010).

A quick review of some of the basic terms used in NLP is required before discussing specific techniques and tools. A corpus is a set of text-based documents, and a corpora is two or more sets of text documents. Syntax is the ruleset for creating a sentence, while semantics has to do with the meaning of a sentence. Semantics requires associating words and concepts and structuring words into phrases (Kapetanios et al. 2013). Context provides additional meaning beyond the structure of a sentence. Tokens are the subsets of sentences and words derived from a corpus. Stopwords need to be identified as they provide little semantic value, and these can be removed. Examples include *a*, *an*, *the*, etc.

NLP has several challenges. For example, polysemy refers to the notion that words have different meaning based on context, which illustrates the richness of natural language. Resolving the differences is an important aspect of NLP. Normalizing words, which includes lemmatization and stemming, is an operation where different forms of words are reduced to the root, or base, word. Another process called part-of-speech tagging is a process where words are tagged as nouns, verbs, adjectives, etc. A similar concept is Named Entities, where specific words or groups or words are tagged based on some features such as proper nouns, addresses, names, etc.

In the business and financial analysis fields, Costantino and others proposed the following process to extract information: morphology, parsing, semantics, and pragmatics (Constantino et al. 1997). A similar but extended information extraction approach was proposed nearly 20 years later by Grishman (2015). He proposed the following steps: named entity tagging, syntactic analysis, internal coreference resolution, semantic analysis, and external coreference analysis. The latter naming convention is more contemporary and introduces the notion of coreference, which is needed when a word is shortened, abbreviated, or substituted by a pronoun.

The concept of information extraction, which Cardie (1997) describes as a system that "summarizes the text with respect to a prespecified topic or domain of interest," relies on these processes. In this approach, and in the case where language is very structured and unembellished, semantic analysis may not be needed. However, as Grishman (2015) points out, other sources of information such as news feeds are free to be more expressive and articulate; therefore, semantic analysis is required.

There are many tools that support NLP, ranging from ones that directly interface with ontology editors such as Protégé (2015) and others built around programming languages such as Python. More relevant to this research is how NLP can be used to provide an ABM characterization through experts' elicitations. Given they describe the phenomenon in their own way and through their unique worldviews, the intent is to process that information in such a way it provides a characterization of a model, and in this case an ABM. The next section describes a tool and process that begins the endeavor of characterizing ABM through semi-automated processes.

4. ABM NLP ANALYZER

The intent of NLP is to capture unstructured narratives of a phenomenon and generate an ABM specification that engages non-simulationists in ABM conceptualization. This is achieved by characterizing a domain in terms of agents, attributes and rules. We consider the agent-attribute-rule characterization basic but a necessary step to understand ABM conceptualization.

The analyzer was developed using IBM Natural Language Understanding (NLU) and Knowledge Studio (KS), both commercially available tools. Readers should be able to follow the paper and replicate this approach. Further, they should be able to apply it to specific domains where the analyzer should generate domain-accurate results.

IBM provides the NLU with built-in capabilities to take a document as an input and identify sentiment, emotion, keywords, entities, and concepts. The results are displayed for each category, and these insights can be used for analytical purposes. The Knowledge studio enables users to create NLP models with custom entities as shown in Figure 1. KS builds a corpus by annotating documents with custom entities that the modeler desires the NLP to identify, which in this case are agent, agent attributes and agent rules.

Figure 1: Agent and attribute entities in IBM KS.

The ABM analyzer ingests an ABM description as input and returns candidate agents, agent attributes, and rules. This output provides an initial ABM specification assisting the user(s) in further developing their conceptual model(s). The process may also elicit additional insight into the model by visualizing the connections within the description.

The guidelines developed by Pustejovsky and Stubbs (1999) are used to create a corpus and develop annotations, which was performed by a single person. While this is a limitation within the current study, it was deemed sufficient given the annotation process was discussed with another member of the team over several iterations evaluating for input correctness and outcome accuracy. The corpus comes from papers reporting ABM implementations that capture a description of a phenomenon. Ten ABM descriptions from the Journal of Artificial Societies and Social Simulation (JASSS) were used for training. Three descriptions were used for testing: one created by the authors to test the initial baseline; one from SpringSim 2019 and one from the Journal of Religion, Brain & Behavior. Lastly, embedded document citations were deleted to avoid introducing noise during training. It is important to note that a large portion of ABM publications do not capture a description of phenomena. Not only did this make obtaining papers for training and testing more difficult, but more importantly, this points to a larger problem: the focus on implementation assuming the reader does not need to understand a phenomenon's larger context.

Through the NLP model creation process, we identified the need for two models instead of one: one that identifies agents and agent attributes and a second model that identifies agent rules only. The second model was created because rules are generally longer strings and often contain an agent or attribute. In early testing, an inverse relationship was discovered: as rule identification became more accurate, agents and attributes became less accurate. This relationship required a separation of rules from agents and attributes.

KS allows the user to include new descriptions and correcting existing annotations to retrain the model. This process is repeated until the model is at the desired accuracy level. It is suggested using three to four documents every iteration of model update. As mentioned before, this model was created to analyze any phenomenon description from any domain, and better results should be generated if applied to a particular domain.

Annotation required basic rules, a protocol, for the authors to identify agents, agent attributes and rules. These rules are mostly syntactical in nature and require further elaboration:

- Agents perform tasks or actions and are often nouns like person, group, animal, etc.
- Attributes are captured by words that point to a variable such increasing/decreasing or representing a degree/level. Attributes are often nouns or adjectives, which make their annotation and identification challenging. Noun examples are age, wage, weight, and productivity. Adjective

examples are hot/warm (representing degree in temperature) or abundant/scarce (representing scale in size).
- Rules often contain action verbs and are usually related to agents or attribute; i.e. *societies with more power will often attack neighbors* or *people who work have an income*.

After training is complete, an accuracy test is performed. Testing is conducted iteratively to identify errors or poor training early. The test papers are compared to the specification generated by the ABM analyzer, and a spreadsheet is used to record and compare the multiple results.

It is important to note:

- An agent or attribute may be referred by multiple different names. The non-simulationist should decide if they refer to the same or different agent/attribute.
- A rule identified by the analyzer should correspond to a rule from the testing paper, if it is captured in the text, to be considered correct.
- Attempts must be made to avoid overfitting, which occurs when the model compliance is highly coupled to the training set, which in this case is unstructured test. This is mitigated through the iterative review process and using wide array of training data.

5. RESULTS

As a test case, a description of the role of elites in social norm diffusion from Salimi et al. (2018) was used to identify agents, agent attributes and rules using the analyzer. The paper has an ABM implementation, which we used for verification purposes. The following is the description provided by the lead author:

Advocates are not always socially positioned to be effective norm promoters on their own. For this reason, they rely on elites in a society who have large networks and strong influence over others such as celebrities and politicians. By convincing elites to adopt norms, advocates increase their ability to reach a wider public audience. It should be noted that advocates promote a norm based on a logic of appropriateness, while elites advertise a norm based on the logic of consequences. In other words, advocates are often not assessing the utilities or benefits of the norm and will never change their mind about that norm; they are like zealots about the norm. Elites, on the other hand, assess the utilities and benefits of norm adoption relative to their level of social power. Iglič and Rus suggest that less powerful, new elites may be targets of norm promoters/advocates because they see the new norm as a potential tool to maximize their utility and balance power against more powerful, old elites. While advocates my use international organizations or nongovernmental organizations to convince elites to adopt norms, and social media also plays a significant role in this process, the current version of the model does not account for these specific means of convincing elites. The tipping point in the norm life cycle is reached when the balance of power between opponent and proponent of the new norm has shifted. At that point, the new norm will begin to be substituted for the existing norm, or it will become the dominant norm. When the norm's life cycle reaches the tipping point, it means the dominant norm is the new norm and a considerable percent of the population supports that norm. After this point, the spread of the norm will reach a higher speed compared to the previous step, and many people from different groups of society may adopt the norm. Norm cascading for humanitarian norms may be different from other norms because degradation of humanitarian values has a diverse and exponential relation to the percent of advocates and therefore is a limiting factor in this process of cascading. This denotes a key difference in norms that have associated punishments, which would prevent individuals from losing interest in a norm over time and delaying the cascade process of norm adoption. During the cascade phase of norm adoption, elites must socialize a norm to persuade others to adopt. Since individuals' motivation to adopt a new norm is based on the logic of consequences—assessing the costs and benefits of adoption for each new norm—it is impossible for elites to socialize norms and persuade others who have no desire to be a part of that social group. Again, this is an important distinction between norms that are adopted through punishment mechanisms. In fact, peer pressure may be regarded as a form *a social punishment by which certain norms are socialized to others and individuals find that socially the benefits of adoption outweigh the costs.*

The test results shown in Table 2 indicate an increase on accuracy over the course of two iterations when compared to the testing paper specification. Comparing the analyzer version 1 results to the paper's specification, the agent names were few and generic only matching "people." Concluding that this was due to lack of variety of agents in the training data, the number of training papers with a larger variety of annotated agents and agent attributes was increased. Version 2 provides a larger number of identified words with a higher correspondence to those specified in the testing paper. However, there are more candidate agents identified. This is where the subject matter expert would clean the list based on the domain of interest. Tables 3 lists the results pertained to rules following the same process.

Table 2: Agent comparison between model versions given successive iterations of reconciliation.

Agents			**Attributes**		
Author Defined	**ABM Analyzer Version 1**	**ABM Analyzer Version 2**	**Author Defined**	**ABM Analyzer Version 1**	**ABM Analyzer Version 2**
elites people advocates	individuals** people* humanitarian**	advocates* zealots** elites* society** promoters** model celebrities** people* individuals**	influence power social norm	norm* social power* politicians groups	consequences** benefits** pressure** costs and benefits** social power* influence * social norm* utilities organizations

* equivalent to testing paper's specification
** deemed similar to testing paper's specification

Table 3: Rules comparison between model versions given successive iterations of reconciliation.

Author-Defined Rules
• Elites can encourage followers to do what they say • Getting elites to advocate for your norm will increase ability to reach more people
ABM Analyzer Version 1
• society who have large networks and strong influence over others such as celebrities and politicians** • advocates promote a norm based on a logic of appropriateness** • humanitarian values has a diverse and exponential relation to the percent of advocates and therefore is a limiting factor in this process of cascading** • social punishment by which certain norms are socialized to others and individuals find that socially the benefits of adoption outweigh the costs** • while elites advertise a norm based on the logic of consequences** • social power • less powerful • new elites may be targets of norm promoters/advocates because they see the new norm as a potential tool to maximize their utility and balance power against more powerful* • old elites

• tipping point in the norm life cycle is reached when the balance of power between opponent and proponent of the new norm has shifted* • it will become the dominant norm • many people from different groups of society may adopt the norm*
ABM Analyzer Version 2
• advocates are not always socially positioned to be effective norm promoters on their own* • convincing elites to adopt norms, advocates increase their ability to reach a wider public audience* • many people from different groups of society may adopt the norm* • norm cascading for humanitarian norms may be different from other norms because degradation of humanitarian values has a diverse and exponential relation to the percent of advocates and therefore is a limiting factor in this process of cascading** • motivation to adopt a new norm is based on the logic of consequences-assessing the costs and benefits of adoption for each new norm-it is impossible for elites to socialize norms and persuade others who have no desire to be a part of that social group** • it should be noted that advocates promote a norm based on a logic of appropriateness, while elites advertise a norm based on the logic of consequences* • advocates are often not assessing the utilities or benefits of the norm and will never change their mind about that norm; they are like zealots about the norm* • suggest that less powerful, new elites may be targets of norm promoters/advocates because they see the new norm as a potential tool to maximize their utility and balance power against more powerful, old elites** • advocates my use international organizations or nongovernmental organizations to convince elites to adopt norms, and social media also plays a significant role in this process, the current version of the model does not account for these specific means of convincing elites* • tipping point in the norm life cycle is reached when the balance of power between opponent and proponent of the new norm has shifted* • at that point, the new norm will begin to be substituted for the existing norm, or it will become the dominant norm* • when the norm's life cycle reaches the tipping point, it means the dominant norm is the new norm and a considerable percent of the population supports that norm** • after this point, the spread of the norm will reach a higher speed compared to the previous step**

* equivalent to testing paper's specification
** deemed similar to testing paper's specification

As in Table 2, one can see there is an improvement in accuracy from version 1 to (two rules equivalent to testing paper and five rules deemed similar) version 2 (six rules equivalent and five rules deemed similar). Yet, more candidate rules were identified requiring further cleaning by the subject matter expert during the conceptualization process.

6. DISCUSSION AND FUTURE WORK

There are three elements to highlight: the importance of a description, the agent-attribute-rule characterization, and the NLP characterization. *Providing a phenomenon description* in ABM publications is often overlooked. A description of a phenomenon, even if brief, provides grounding, either coming from data, theory or both. This grounding facilitates the understanding not only of the phenomenon but also of the model representing the phenomenon. Model specifications are common taking the form of artifacts such as activity diagrams or flow charts. However, these artifacts focus on specifying the ABM implementation and do not convey the context of the phenomenon from which the implementation is derived. These practices may create barriers to entry for professionals or scholars that are not familiar with modeling programming practices, as they cannot reconstruct the phenomenon narrative from a model description. Providing a narrative of a phenomenon without ABM jargon should be a normal practice. The narrative in this case should be implementation independent.

A description of the phenomenon, followed by an *agent-attribute-rule characterization*, provides traceability to non-simulationists; the means of connecting a narrative to an ABM specification. This characterization *does not* require a model constructed from NLP. It can be done using pen and paper, post-it notes or any other means that facilitate the elicitation, discussion, and the formulation of the phenomenon of interest. This simple characterization is a bridge between subject matter experts and modelers that can be further elaborated with a list of assumptions and coding artifacts, which is needed for documentation. Figure 2 illustrated a post-it notes representation of the Schelling Segregation model NetLogo code as an example.

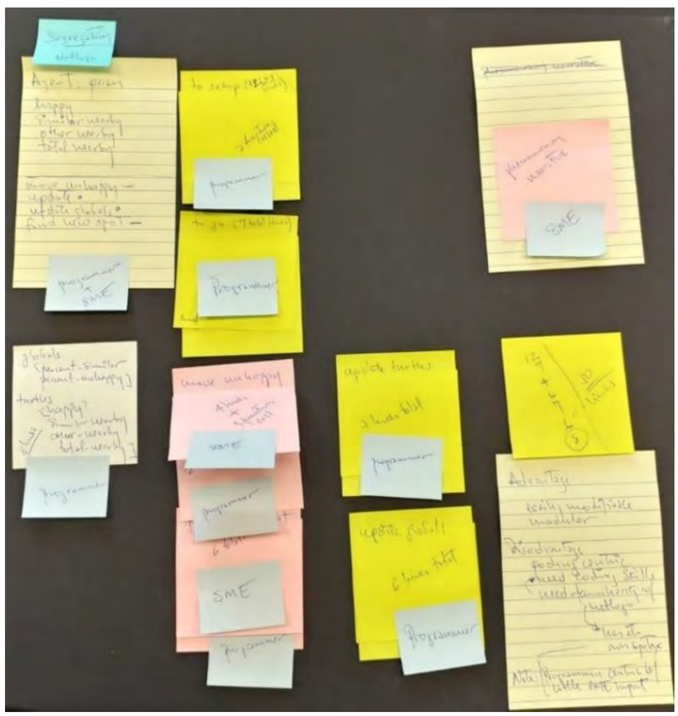

Figure 2: Post-it notes of NetLogo code.

This simple but effective representation captures information that relates to the phenomenon (pink notes) and information that is related to code (bright yellow notes). Pink notes capture agent behaviors, which in this case refer to movement and conditions that cause movement. The agent, or turtle, moves when it is unhappy and prefers a neighbor type location. Yellow notes refer to expressions such as updating the turtle variables. It is noted that this is not an agent-attribute-rule representation. It is a post-it notes representation reverse engineered from the NetLogo code. It is meant to illustrate how knowledge could be elicited from the subject matter experts, and it facilitates the explanation to non-simulationists of their potential role in the modeling process. It is expected that the expert in particular domains such as agent movement given a set of conditions may not be a simulation expert, and so eliciting their knowledge in such a way that is natural to that information to describe the agent-attribute-rule characterization. The expert should be able to provide knowledge in an organic way.

Simple ABM characterization may facilitate the creation of tools that do not rely on developing computer programs or software. Figure 3 shows a tool (under development) that implements ABMs using an agent-attribute-rule-expression characterization. Rules, like mentioned before, either capture a phenomenon's candidate rules or a coding component. Expressions in this case are coding components that return a value.

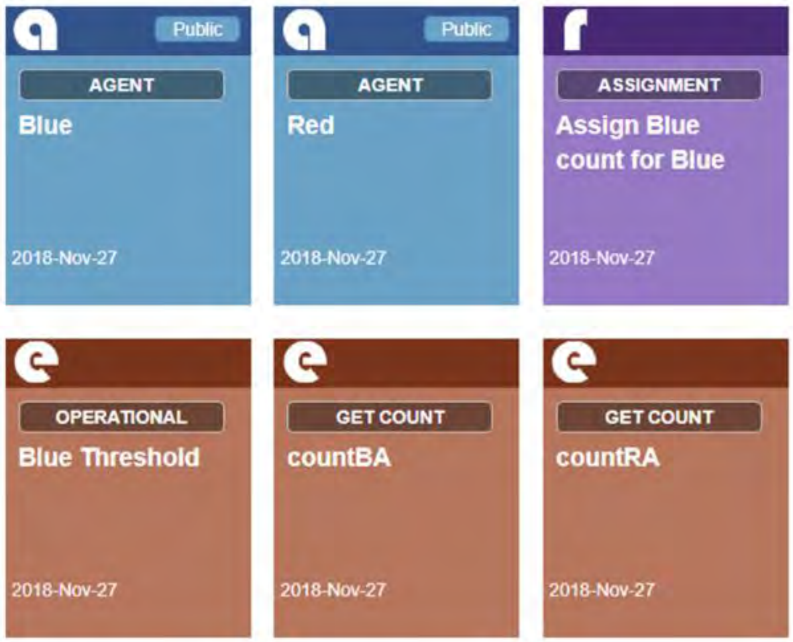

Figure 3: ABM tool implementing an Agent-Attribute-Rule-Expression characterization.

New tools, following a manual or semi-automatic NLP-based ABM characterization could further bridge the gap between subject matter experts and modelers by enabling the former to participate effectively in the modeling process. With the subject matter experts supporting the modeling process, valuable knowledge is elicited, which also facilitates the validation process.

Lastly, and on the topic of discussion, an *NLP-based ABM characterization* has the potential for not only facilitating the ABM characterization for non-simulationists but also supports the development of new tool capabilities for the development of ABMs. The IBM Watson platform has advantages like ease of use and an API for web accessibility, potentially bringing ABM NLP to many. Exposing the ability to create ABM from NLP to the non-simulationist is a stepping off point for further development. The platform has limitations, which can be overcome with more specialized tools. Perhaps more interestingly, it is the capability of not only characterizing ABMs but also searching ABMs using characterizations such as the agent-attribute-rule. Furthermore, since the data model is formalized and captured in a standard manner, reasoning may be performed across models for additional insight into model development from exogenous queries. Also, as additional models are added, it is expected to increase the richness of semantic availability.

Such specialized tools will rely on adding more semantic information to the model. Provided the data model, visualizing the graph through a series of inter-connected nodes and edges will provide another perspective designed to note only externalize the model but also elicit more knowledge. It is intended that a directed graph be editable while maintain consistency with the model. This interactive approach and its benefits to constructing qualitative models through concept mapping, decision tree analysis, and scenario building is referenced by Voinov and others (2018). Another perspective was taken by Pillutla and Giabbanelli who combined fuzzy cognitive maps with NLP and developed an interactive mechanism to explore relationships to aid in the development of concept maps from natural language (2019).

Related concepts were captured by Gupta and others (2018) where conceptual models were used to externalize varying terms intended to mean the same concept and then align the concepts across views. A web search using the phrase "fore score and seven years ago" will return results based on "four score and seven years ago." The semantic understanding based on the context trumps the syntax. To put into the context of a well-known agent-based model, predator-prey, the non-specialized modeler may intend to describe the relationship between foxes and rabbits by stating "foxes eat rabbits." A specialize NLP tool will associate the word "eat" with "consume," by leveraging a broader corpus of information. A pattern begins to develop based on this simple association, and the tool begins to aid the modeler by suggesting other agent rules such as "rabbits reproduce," which the modeler can accept or reject based on the intent of the study's specific purpose.

Lastly, generating an ABM specification towards facilitating conceptualization goes beyond the identification of agents, agent attributes, and rules. One crucial aspect is matching those three aspects to a modeling question. Even so, a generated specification may be over or under specified. Yet, we hypothesize that this approach facilitates engagement and learning by non-simulationists. We expect to evaluate this hypothesis empirically in the future. Anecdotally, we used the approach manually (identifying agents, agent attributes, and rules from a phenomenon narrative and mapping that to an implementation) with a group of scholars in the humanities. Conveying the approach and the resulting implementation was seamless as we use their language, not the modeler's language to explain the phenomenon conceptualization and implementation.

ACKNOWLEDGEMENTS

This material, in part, is based on research sponsored by the Office of the Assistant Secretary of Defense for Research and Engineering (OASD(R&E)) under agreement number FAB750-15-2-0120. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Office of the Assistant Secretary of Defense for Research and Engineering (OASD(R&E)) or the U.S. Government.

REFERENCES

Bersini, H. 2012. "UML for ABM". *Journal of Artificial Societies and Social Simulation* 15(1):1–9.

Cardie, C. 1997. "Empirical Methods in Information Extraction". *AI Magazine* 18(4):65–79.

Costantino, M., R. Morgan, R. Collingham, and R. Carigliano. 1997. "Natural Language Processing and Information Extraction: Qualitative Analyis of Financial News Articles". In *Proceedings of the IEEE/IAFE Computational Intelligence for Financial Engineering*, March 24th–25th, Piscataway, New Jersey, 116–122.

Epstein, J. 2008. "Why Model". *Journal of Artificial Societies and Social Simulation* 11(4):1–12.

Ghorbani, A., P. Bots, V. Dignum, and G. Dijkema. 2013. "MAIA: A Framework for Developing Agent-Based Social Simulations". *Journal of Artificial Societies and Social Simulation* 16(2):1–19.

Grimm, V., U. Berger, D. DeAngelis, J. Polhill, J. Giske, and S. Railsback. 2010. "The ODD protocol: A Review and First Update". *Ecological Modelling* 221(23):2760–2768.

Grimm, V., G. Polhill, and J. Touza. 2013. "Documenting Social Simulation Models: The ODD Protocol as a Standard". In *Simulating Social Complexity: A Handbook*, edited by B. Edmonds and R. Meyer, 117–133. Berlin: Springer.

Grishman, R. 2015. "Information Extraction". *IEEE Intelligent Systems* 30(5):8–15.

Gupta, V. K., P. J. Giabbanelli, and A. A. Tawfik. 2018. "An Online Environment to Compare Students' and Expert Solutions to Ill-Structured Problems". In *Learning and Collaboration Technologies. Learning and Teaching*, edited by P. Zaphiris and A. Ioannou, 286–307. Cham: Springer.

Kapetanios, E., D. Tatar, and C. Sacarea. 2013. *Natual Language Processing: Semantic Aspects*. 1st ed. Boca Raton: CRC Press.

Manning, C. and H. Schutze. 1999. *Foundations of Statistical Natural Language Processing*. 1st ed. Cambridge: MIT Press.

Martinez, A. 2010. "Natural Language Processing". *WIREs Computational Statistics* 2(3):352–357.

Musen, M. A. and Protégé Team. 2015. "The Protégé Project: A Look Back and a Look Forward". *AI Matters* 1(4):4–12.

Padilla J. J., E. Frydenlund, H. Wallewik, and H. Haaland. 2018. "Model Co-creation from a Modeler's Perspective: Lessons Learned from the Collaboration Between Ethnographers and Modelers". In *Social, Cultural, and Behavioral Modeling*, edited by R. Thomson, C. Dancy, A. Hyder, and H. Bisgin, 70–75. Cham: Springer.

Pillutla, V. S. and P. J. Giabbanelli. 2019. "Iterative Generation of Insight from Text Collections through Mutually Reinforcing Visualizations and Fuzzy Cognitive Maps". *Applied Soft Computing Journal* 76:459–472.

Pustejovsky, J. and A. Stubbs. 2013. *Natural Language Annotation for Machine Learning*. 1st ed. Sebastopl: O'Reilly Media.

Reinhardt, O., A. Ruscheinski, and A. M. Uhrmacher. 2018. "ODD+P: Complementing the ODD Protocol with Provenance Information". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 727–738. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Salimi, K., E. Frydenlund, J. J. Padilla, H. Haaland, and H. Wallevik. 2019. "The Role of Elites in the Diffusion of Social Norms of Humanitarianism". In *Proceedings of the 2019 Annual Simulation Symposium,* edited by A. A. Del Barrio, C. J. Lynch, F. J. Barros, X. Hu, and A. D'Ambrogio, 1–12. San Diego: Society for Computer Simulation International.

Siebers, P. and B. S. S. Onggo. 2014. "Graphical Representation of Agent-Based Models in Operational Research and Management Science Using UML". In *Proceedings of the Operational Research Society Simulation Workshop*, edited by B. S. Onggo, C. Heavey, T. Monks, B. Tjahjono and D. J. van der Zee, 143–153. Birmingham, UK: Operational Research Society.

Socher, R., A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, and C. Potts. 2013. "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank". In *Proceedings of Empirical Methods in Natural Language Processing,* edited by D. Yarowsky, T. Baldwin, A. Korhonen, K. Livescu, S. Bethard, 1631–1642. Seattle: Association for Computational Linguistics.

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.

Voinov, A., K. Jenni, S. Gray, N. Kolagani, P. D. Glynn, P. Bommel, C. Prell, M. Zellner, M. Paolisso, R. Jordan, E. Sterling, L. S. Olabisi, P. J. Giabbanelli, Z. Sun, C. Le Page, S. El Sawah, T. K. BenDor, K. Hubacek, B. K. Laursens, A. Jettert, L. Basco-Carrerau, A. Singer, L. Young, J. Brunacini, and A. Smajgl. 2018. "Tools and Methods in Participatory Modeling: Selecting the Right Tool for the Job". *Environmental Modelling and Software* 109:232–255.

AUTHOR BIOGRAPHIES

JOSE J. 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.

DAVID SHUTTLEWORTH is a doctoral student at the Modeling, Simulation, and Visualization Engineering Department at Old Dominion University. He is a member of the Department of Navy's Modeling and Simulation Office and specializes in warfare analysis, multi-objective optimization, and model-based engineering. His research interests are in semi-automated methods to constructing models. His email address is dshut002@odu.edu.

KEVIN O'BRIEN is an undergraduate student at the Computer Science Department at Old Dominion University. He has experience working with natural language processing and is focusing on learning new methods of machine learning at Virginia Modeling Analytics Simulation Center as an intern. His email address is kobri001@odu.edu.