GE Global Research

DARPA-PA-18-02
DARPA PA for AIE
Volume 1: Technical and Management Volume

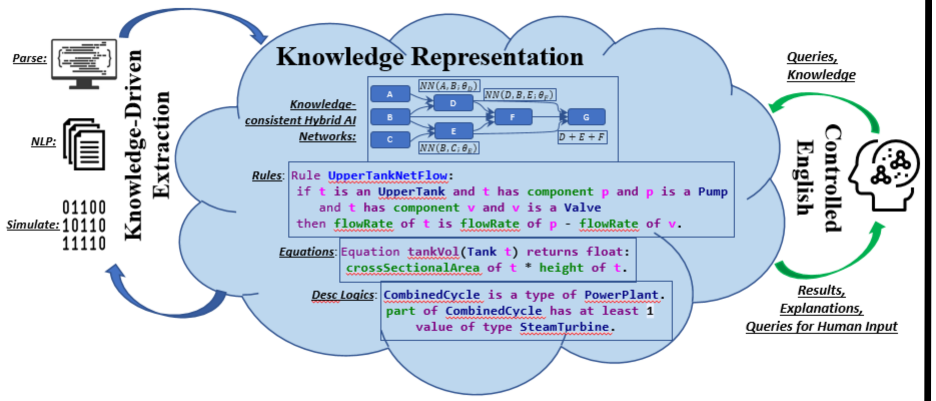
AIE Opportunity #	DARPA-PA-18-02-01 (TA1)
Proposal Title	**ANSWER**: **A**ugmented Bayesian **N**etworks Integrating **S**emantics **W**ith **E**xtraction and **R**eadability
Proposer Organization	General Electric Company, GE Global Research
Type of Organization	Large Business
Proposer's Internal Reference Number, if any	GEGR-18-273
Technical Point of Contact (POC)	Dr. Andrew Crapo GE Global Research One Research Circle Niskayuna, NY 12309-1027 (518) 387-5729 crapo@ge.com
Administrative POC	Ms. Marlee Rust GE Global Research One Research Circle Niskayuna, NY 12309-1027 (518) 387-4252 rust@ge.com
Award Instrument Requested	Other Transaction
Total Proposed Price (not to exceed $1,000,000)	Phase 1: $299,873 (customer share $199,978) Phase 2: $699,817 (Customer share $466,603) Total: $999,689 (customer share ($666,581)
Place(s) of Performance	Niskayuna, New York
Other Team Members (subawardees and consultants), if any	None
Date Proposal was Prepared	September 17, 2018
Proposal Validity Period (minimum 365 days)	September 17, 2019

Augmented Bayesian **N**etworks Integrating **S**emantics **W**ith **E**xtraction and **R**eadability (ANSWER)

GE Global Research; Andrew W Crapo, PhD

DARPA-PA-18-02
Artificial Intelligence Exploration (AIE)

CONCEPT



```
Rule UpperTankNetFlow:
 if t is an UpperTank and t has component p and p is a Pump
    and t has component v and v is a Valve
 then flowRate of t is flowRate of p - flowRate of v.
```

```
Equation tankVol(Tank t) returns float:
 crossSectionalArea of t * height of t.
```

```
CombinedCycle is a type of PowerPlant.
part of CombinedCycle has at least 1
    value of type SteamTurbine.
```

APPROACH

- Capture scientific knowledge in a modular Bayesian architecture that combines Description Logics-style semantics capturing domain knowledge with equations for physics-based models and constraints and neural networks for data-driven models and imprecise knowledge including causality.
- Create a novel local-global Bayesian inference over all types of knowledge to enable representation accuracy, efficient convergence, task-specific performance, and reusability.
- Extract scientific models from code via parse tree analysis and simulation and from documentation and publications via custom named entity recognition and application of domain knowledge .
- Use controlled-English grammars to facilitate query formulation, make knowledge readable and explorable, and explain results of inference and model evaluation and execution.

IMPACT

Need: To understand and extract scientific knowledge, expressed in code, documentation, or publications, requires deep contextual knowledge ranging from domain concepts to the laws of physics to the syntax and semantics of the source. Extracted knowledge is of various types but must be usable as an integrated whole with inference and querying capability yielding results easily readable and understandable by humans.

Goal: Significantly advance the state-of-the-art by 1) combining disparate knowledge types into a single cohesive Bayesian knowledge network capable of understanding and using scientific models, 2) extracting scientific models from code and text via a knowledge-driven and focused approach, and 3) enabling high-bandwidth bidirectional communication and collaboration between a knowledgeable AI and a human.

CONTEXT

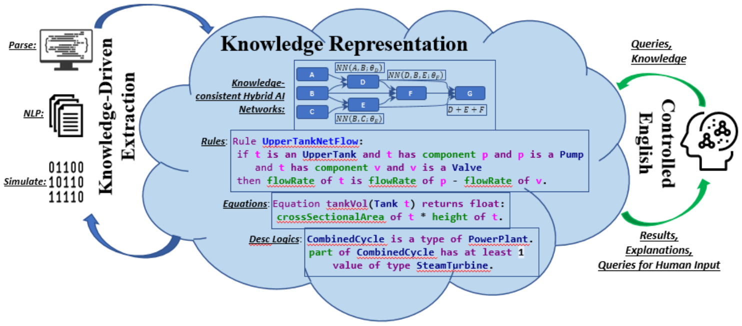
Current approaches:

- Graphical models make use of dependencies but cannot depict exact equations and constraints.
- Bayesian calibration with physics equation and experimental data can learn parameters and capture unmodeled dynamics but do not integrate knowledge of causality and constraints.
- Code analysis tools focus on software errors and vulnerabilities and do not maximally exploit parse trees to extract high-fidelity scientific models.
- NLP is used to do named entity extraction but is not sufficiently informed by domain knowledge and unique “fingerprints” of in-text equations and scientific knowledge.

Table of Contents

1. Proposal Summary

GE Global Research proposes **A**ugmented Bayesian **N**etworks Integrating **S**emantics **W**ith **E**xtraction and **R**eadability (**ANSWER**), a two-phase program to develop and demonstrate 1) knowledge-driven extraction of scientific models from code and texts, 2) a novel and powerful representation of the accumulated models and contextual knowledge, and 3) a controlled-English interface to the knowledge repository facilitating exploration, understanding, execution of models, inference, and explanation of execution/inference results.



```
Rule UpperTankNetFlow:
 if t is an UpperTank and t has component p and p is a Pump
    and t has component v and v is a Valve
 then flowRate of t is flowRate of p - flowRate of v.
```

```
Equation tankVol(Tank t) returns float:
 crossSectionalArea of t * height of t.
```

```
CombinedCycle is a type of PowerPlant.
part of CombinedCycle has at least 1
    value of type SteamTurbine.
```

Figure 1: ANSWER architecture with knowledge-driven extraction and a controlled-English human-computer interface

Discussion

Despite significant advances, artificial intelligence today remains significantly siloed. Traditional knowledge of physics and engineering remains in the realm of text books, publications, and legacy code. Certain kinds of knowledge and data is made more accessible and useful through graph models such as Linked Open Data, but these models remain largely deterministic. Unprecedented collection of observational data has resulted in a burgeoning of machine learning (ML), but ML models remain relatively isolated from a knowledge of the science behind the phenomena. Perhaps the biggest barrier of all is between many humans and knowledge which is captured in anything but natural language.

GE Global Research proposes to tear down barriers by 1) integrating multiple types of knowledge in a novel Bayesian architecture called Knowledge-Consistent Hybrid AI Network (K-CHAIN), 2) use this knowledge to drive better extraction of additional knowledge from scientific model code, documentation, and publications, and 3) use and extend our capabilities in the use of controlled English to make the accumulated scientific knowledge more accessible to people working in a domain. The ANSWER architecture is shown in Figure 1.

There are several technical challenges to be solved in accomplishing this work. One lies in the integration of Description Logics-style semantics and probabilistic representations. Our approach is to use local Bayesian networks to represent probabilistic sub models, which can be refined and eventually replaced with more physics-based models as the science is better understood. This will require a new kind of inference over the disparate representations to achieve an integrated inference and query-answering capability. Another challenge is to make NLP more model/knowledge driven. The state of the art is to extract named entities, then look for corresponding concepts in an ontology or lexical database, and then try to make sense of the phrase, the sentence, and the paragraph. To extract more precise scientific knowledge we must, as humans naturally do, be guided by an awareness of the domain and an expectation of what knowledge we might find. The challenge of extraction of scientific models from code is perhaps more confined. Parse trees for a given language contain all the information but a fair amount of invention will be required to figure out how to efficiently use this information to construct scientific models in the target representation. When building data-driven models via code execution, the challenge will be to automatically execute an existing code with appropriate inputs to build a supervised learning model. Finally, our work in controlled-English interfaces to semantic models has focused on use as a human input mechanism. For this work we will need to generate controlled English from the knowledge store as an output.

The result of successful execution of this project will be a new approach to the representation of scientific knowledge, a new way to add additional information as scientific research extends human knowledge, and a friendlier way of making this knowledge available to modelers, researchers, and those tasked with keeping our military technologically at the leading edge. We propose to spend $299,873 during a 6-month Phase 1 feasibility study followed by a $699,817 Phase 2 proof of concept for a project total of $999,689. Work will proceed on four fronts in parallel: 1) K-CHAIN knowledge representation and computational model generation, 2) knowledge-driven extraction from code, 3) knowledge-driven extraction from text, and 4) controlled-English human/AI interface to hybrid knowledge. We expect spending to be at an approximately constant rate over the period of performance.

2. Goals and Impact

Computer code expressing scientific models contains very high-fidelity knowledge about science. However, it is not generally possible to understand the encoded model without understanding a great deal of implicit context which existed in the mind (mental models) and resources used by the programmer or scientist who authored the code [1]. While some of this context may have been captured in in-line comments or in related documentation, it is never entirely captured and unfortunately is often not captured at all. To accurately extract the scientific knowledge embedded in code, we must regenerate all or part of the context. Fortunately, this context is usually to be found in the general body of scientific knowledge of the domain and is represented in texts, publications, etc. Therefore, it is our goal to integrate the automated extraction of scientific models from code with the construction and use of rich semantic domain models extracted from unstructured and semi-structured sources.

To be maximally useful, the knowledge captured from all sources must be represented in a form supporting many kinds of knowledge, highly understandable by humans, and easily lending itself to various kinds of inference. Finding the state-of-the-art representations to be lacking in one or more of these respects, we have started the development of Knowledge-Consistent Hybrid AI Network (K-CHAIN), a Bayesian architecture which combines the strength of graphical models to depict causal dependencies, structural equations to depict exact information, and a semantic language to represent domain knowledge. Moreover, the architecture can programmatically be converted to code for simulation and inference.

The proposed work will result in a demonstrated methodology to accomplish model-driven extraction of scientific models and related knowledge from code and from texts, a novel and powerful representation of the accumulated models and contextual knowledge, and a controlled-English interface to the knowledge repository facilitating exploration, understanding, execution of models, inference, and explanation of execution/inference results. This will significantly move forward the state-of-the-art in areas of concern to TA1 of DARPA ASKE and will be valuable to the US military in ensuring that their platforms and systems are based on robust and up-to-date scientific understanding.

3. Technical Plan

3.1 Domain of Research and Demonstration

To ground our research in a scientific domain where we have ready access to open source code, documentation, publications, and subject matter expertise, we have chosen NASA's Hypersonic Aerodynamics web site as a source [2]. The site provides a number of simulators that allow interaction with the science through downloadable applets with code implementing a number of the scientific models. Here we also find both textual explanations of the science behind hypersonics (thermodynamics, phases of matter, laws of motion, Navier-Stokes equations, Euler equations, compressible flow, shock waves, etc.) and the various type of propulsion accomplished through application of this science (ramjets, scramjets, rockets, gas turbines). Moreover, hypersonics is an active area of research, which is critical to national security, and several publication houses that allow text mining, such as Elsevier, publish new articles on the subject regularly. Narendra Joshi, Chief Scientist here at GE Research and a world-renown expert in propulsion, is a key person and will ensure the quality of our contextual and code-extracted knowledge. The developed methodology will be agnostic to the domain as much as possible and hence allow easy adaptation to other areas of interest to DARPA and DoD.

3.2 Extraction of Scientific Models from Code

A primary source of scientific knowledge envisioned by ASKE is code instantiating scientific models in a computable manner. Science models can be placed in two general categories. Physics-based models instantiate the equations of a "law" of physics in an executable form. Data models, on the other hand, are based on observation and generated from inputs and associated outputs which are used to train a machine-learning model, usually when the underlying physics is not yet well understood but sometimes for other reasons. In GE we frequently create data-driven models or hybrid models from physics-based models to improve the performance of complex models.

There are two ways of extracting knowledge from code. One can examine the code statically and extract the concepts, relationships, constraints, etc., from the code or one can exercise the code, perturbing the inputs and observing the outputs. Considerable work has been done in both static and dynamic analysis. However, most of this work has been to the end of validating the code; looking for evidence of errors or vulnerabilities. Exceptions to this are tools that show the overall structure of the code, the flow of data through the code, and the control flow of the code. These analyses are typically used in reverse-engineering efforts and may include the extraction of requirements from code. Often UML class diagrams are an output of static analysis. Analysis tools such as IBM Rhapsody allow querying for additional information.

For the purpose of extracting scientific models from code, a more detailed ingestion of the software is needed. One promising approach is to parse the code in the same way that a compiler might do so and analyze the contents of the parse tree. For example, an Xtext grammar and associated parser/lexer for Java would not only generate a parse tree for the code but would capture scoping and hyperlinking of resources [3]. This allows one to easily identify where a variable is defined and all the places where it is referenced. These references will be in parse tree objects that are identifiable as, for example, mathematical operations that involve other variables. This allows the relationships in the code to be made explicit in terms that are easily translated into the semantic domain. For example, an applet to compute thrust from the NASA Hypersonics web site contains this Java code snippet.

```
        npr = 1.0 /prat ;
        uexm = Math.sqrt(2.0*rgas/fac1*tt*(1.0-Math.pow(1.0/nprm,fac1)));
        fgros = mflow * uex / g0 + (pexit - pamb) * athrust * 144. ;
```

Analysis of the parse tree provides information like the following examples:

- npr is only used in a method to generate output
- rgas is defined elsewhere by rgas = runiv * g0 / mweight where g0 = 32.2
- uexm is defined here but never referenced (inference might mark this model as suspect)
- uex is defined in 5 different ways depending upon noztype (convergent nozzle or convergent-divergent nozzle, from in-line comment) and upon machth (sonic flow, choked flow) for convergent nozzle, and upon a number of other variables for convergent-divergent nozzles.

Concepts encountered in the code, e.g., *convergent nozzle*, can be found in a domain ontology or searched for in documentation and literature and then added to the ontology. This approach is similar to that taken to extract semantic models from the UxAS C++ codebase in GE's successful participation in the AFRL 2017 Summer of Innovation [4] [5].

Extraction of knowledge via static analysis can be augmented by extraction via dynamic analysis if an executable version of the code is available or can be generated. If the code is not parsable for some reason, as might be the case if it were only available in binary form or if it were in a language for which no parser is available, then dynamic analysis may be the only means of model extraction. When this is the case, the code can be executed while varying inputs over their range of possible values and observing the outputs. Such a simulation can create training data from which a data-driven model can be generated. Active learning or sequential sampling-based data-driven modeling with the unparsable code treated as a simulator can improve the efficiency and quality of models thus generated. In some cases, it may be possible to analyze the data-driven model to understand main and secondary effects and identify the one or more physics-based equations implicit in the data. In other cases the data-driven model may be opaque, such as when the fit is accomplished with a Bayesian neural network. Even such a model can still be useful because it remains an executable part of the knowledge graph of the scientific domain. Moreover, the uncertainty from imperfect information, due to inability to extract exact relationships from code, is represented in the proposed Bayesian framework. This uncertainty in turn will drive the search of code, documentation, and literature of the domain to refine the computational model with exact models.

3.3 Extraction of Knowledge from Texts

Models extracted from code are to be augmented by extracting models from software documentation and publications. Using text mining to extract such knowledge is an area in which we have experience and are engaged in active research. Scientific models in text (documentation, research publications, etc.) are largely described in the form of mathematical equations. Extracting scientific models from text amounts to extracting the model's equations and their associated parameters. Existing research in parsing scientific text has been largely restricted to identifying topics and key phrases [6]. Very little attention has been devoted to understanding and extracting scientific equations [7] [8]. Equations in text can appear in “raw” form (e.g. *f=m*a*) or can appear as a description in text (e.g. “*force equals mass times acceleration*”). To interpret the raw equations that appear in text, clues can be obtained from the surrounding text. For instance, text around the equations will often expand upon the abbreviations used in the equation (e.g. *force (f)*, *mass (a)*).

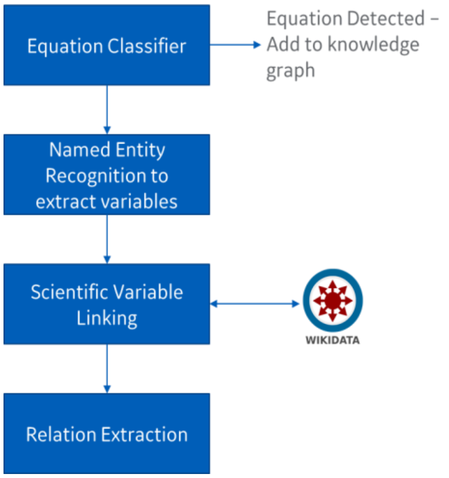

Equation Classifier
Equation Detected – Add to knowledge graph
Named Entity Recognition to extract variables
Scientific Variable Linking
WIKIDATA
Relation Extraction

Figure 2 Our proposed approach extracts not only the direct equation, but also the equations mentioned in text

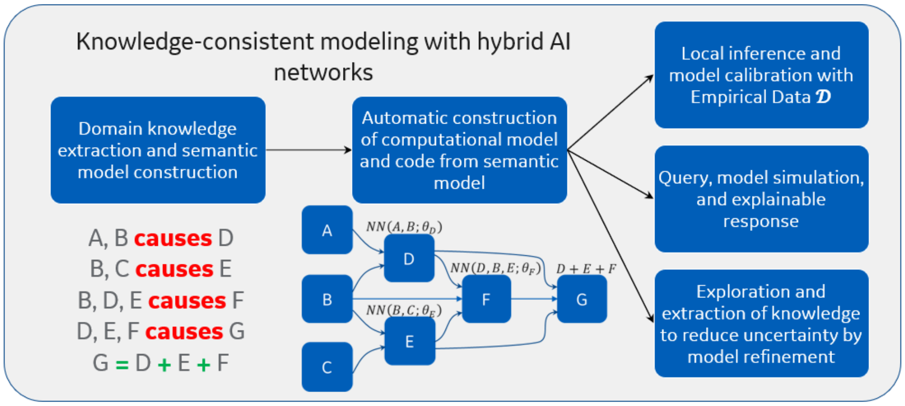
Extracting equations from text thus requires the ability to identify the variables of an equation and the relationships between them and detect if a raw equation appears directly in the text. Figure 2 gives an overview of our approach to accomplish these tasks. We begin by breaking the text into sentences and classifying each sentence to determine if it contains a raw equation. We will train a supervised classifier based on features such as number of tokens in a sentence, length of each token, presence of mathematical symbols or special characters and so on to achieve this task. Equations in text typically consist of less tokens of smaller length and have a large proportion of special symbols as opposed to normal text. Detected equations will be added to an instance of the scientific model in our knowledge graph. Further, a regular expression-driven technique will be used to locate the expanded abbreviations in the surrounding text.

Texts that are not classified as equations will be further processed. First, we will use a custom named entity recognition (NER) system to tag and extract scientific terms and variables mentioned in text. Examples of such mentions include *mass flow rate*, *exit velocity*, *density* and so on. We will leverage a hybrid architecture that combines a deep neural network with a conditional random field for our NER system as proposed in [9]. Training data will be bootstrapped by leveraging tagged documents in NASA’s Hypersonic Aerodynamics Index. Additional training data can be generated by identifying a list of common scientific terms for a given domain from Wikipedia and using a dictionary-based approach such as UIMA ConceptMapper to automatically annotate scientific documents [10] We will further enrich our semantic scientific model by linking these variables to existing entities in knowledge graphs such as Wikidata or DBpedia (e.g. linking *acceleration* to https://www.wikidata.org/wiki/Q11376). This will automatically provide us additional context such as dimensions, symbols, units, etc. for the variable. We propose to use existing entity linking techniques based on syntactic and semantic similarity between the labels of the variables.

Entity extraction and linking will be followed by classification of relations between variables. In cases where multiple variables are extracted from the same sentence, we will leverage a supervised bootstrapping approach to identify the relationship between the variables. Specifically, we will focus on two types – (a) depends (e.g. *thrust* **depends** on *mass flow rate*, *exit velocity*) and (b) equation description (*force* is **equal** to *mass flow rate* **times** *change in velocity*). For the latter, hand-crafted rules will be used to convert variables and relations into raw equations. There are a limited number of ways in which mathematical operations are described using words – we will capture these patterns as rules in our semantic model. While such rules can be inferred or learned, we will start with a hand-crafted approach to bootstrap the system. For the relation extraction and classification task, we will leverage our experience in developing a user-interactive relation extraction system, which learns to extract relations from text with limited labelled data from domain experts [11]. Note that class and property hierarchies in the semantic knowledge allows us to combine sources in sensible ways. Since *thrust* is a subclass of *force*, and *equal* means that the left hand side is *directly proportional* to terms in the right hand side, and *directly proportional* is a sub-property of *depends*, we know that the information in (a) is not different from that in (b), but rather (b) is a more specific statement encompassing (a). Without this type of partial ordering of equations as well as classes and properties, any knowledge representation would become chaotic.

3.4 Knowledge Representation and Reasoning

While extracting knowledge from code is an enabler, it is not sufficient. The extracted knowledge must be represented in a way that lends itself to inspection, querying, analysis, extension, execution, and inference of new knowledge. We propose to meet these requirements with a novel architecture called Knowledge-Consistent Hybrid AI Network (K-CHAIN). K-CHAIN represents disparate scientific knowledge to provide a physics-aware and causality-consistent model of complex systems. We combine the strength of graphical models, structural equations, and semantic languages to represent domain knowledge in a Bayesian architecture that captures all available knowledge including causality and other relational information, symmetry, equations, statistical models, physical constraints, system topology, and conservation laws.


Knowledge-consistent modeling with hybrid AI networks

Domain knowledge extraction and semantic model construction

Automatic construction of computational model and code from semantic model

Local inference and model calibration with Empirical Data $\mathcal{D}$

Query, model simulation, and explainable response

Exploration and extraction of knowledge to reduce uncertainty by model refinement

A, B **causes** D
B, C **causes** E
B, D, E **causes** F
D, E, F **causes** G
G **=** D **+** E **+** F

Figure 3: Overview of K-CHAIN with an example combined cycle power plant

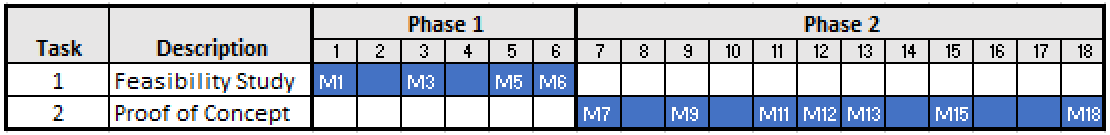
To illustrate, consider a toy example. Figure 3 shows-a simplified combined cycle power plant with two gas turbines and one steam turbine. Here variables *D* and *E* denote power generated by the gas turbines at base load and variable *F* represents power generated by the steam turbine, all of which depend on ambient conditions (variable *B*). Gas turbine power also depends on control inputs (variable *A* and C). Relations like "*D depends on A,B*" or "*A, B causes D*" are commonly encountered in engineering due to an unknown physical relation with empirical knowledge or because the physics is too complicated for the application at hand. To use this knowledge, we postulate that a conditional probability density $p(D|A, B)$ can capture the cause-effect relationship. However, the structure of this density is unknown. At the other extreme, we know deterministically that $p(G|D, E, F) = D + E + F$, where *G* is total power, so the structure is fully known. To allow nonlinear interactions to be modeled, we will leverage the universal approximation property of neural networks and say $p(D|A, B) = p(NN(A,B;\ \theta_D)|A,B)$ with suitable prior over parameters(θ_D). Unlike a pure data-driven model, the above is consistent with cause-effect relations and avoids learning spurious correlations with other variables, e.g., *C*. Thus, a generative model for each vertex, which is not a leaf node, conditioned on its in-neighbors is represented as a Bayesian neural network due to the lack of a known structure. Physics equations and causal relations, now represented by equations, can be put in a unified structure of a computational graph, which can be programmatically created from knowledge graphs in a specific programming language, such as in Python by using TensorFlow or Keras. This modular generative architecture can perform local operations of refinement such as appending and pruning as more scientific knowledge becomes available. Changes can immediately be reflected in the corresponding code for simulation and inference. The generative model enables hypothesis generation and curiosity in the AI to actively seek information and human interaction, which further reduces its uncertainty in the curated knowledge. The crucial challenge is to infer the right computational representation of a semantic relation discovered from text as priors over parameters or variables, structure of the computational graph, and conditional likelihood models. A few common relations, such as causality, will be preprogrammed. The system will learn representations of other relations by exploration and knowledge refinement and by human interaction.

3.5 Knowledge Accessibility by Humans

While our approach goes beyond traditional knowledge graph technology to adequately represent scientific knowledge, it still relies on semantic models of the domain to define the concepts forming nodes in the computational graph. These models are key to all interaction with humans and must match their mental models. We have invested years of research in making semantic models more transparent and usable. This work has given rise to two open source projects. The Semantic Application Design Language (SADL) provides a controlled-English grammar and a rich integrated development environment for Web Ontology Language (OWL) ontologies plus rules and equations as well as tools for model validation and maintenance [12] [13]. The Semantics Toolkit (SemTK) provides an integrated platform for user-friendly querying and semantic data management [14]. In addition, we have extended the controlled-English of SADL to create a SADL Requirements Language (SRL) which is the frontend of the ASSERT™ requirements capture, analysis, and test case generation tool suite [15] [16]. Our work in the 2017 AFRL Summer of Innovation demonstrated both the utility of SRL as a formal requirements modeling language and the ability to extract software requirements from legacy code. We expect controlled-English grammars to be a key element in making scientific knowledge extracted from code, documentation, and publications understandable by humans, in providing explanation and query response, and in enabling the human-computer collaboration that will be necessary to curate extracted knowledge in the foreseeable future.

3.6 Schedule and Milestones

Figure 4 shows the main tasks and milestones over the course of the proposed 18 month project. More detailed subtasks and a description of the milestones are shown in Table 2.



Task	Description	Phase 1						Phase 2											
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	Feasibility Study	M1	■	M3	■	M5	M6												
2	Proof of Concept							M7	■	M9	■	M11	M12	M13	■	M15	■	■	M18

Figure 4: Proposed program schedule

4. Capabilities/Management Plan

Table 1 shows key individuals, their project roles, and the expertise which they bring to the proposed work. Full resumes are available [17].

Table 1: Key personnel roles and expertise

Key Individual	Project Role	Capabilities and Expertise
Andrew Crapo (PI) Principle Scientist PhD Decision Sciences and Engineering System, RPI	Semantic modeling, extraction from code, project oversight	Over 37 years of experience modeling diverse cyber and physical systems from energy conversion to cyber security and including human-computer interfaces and interactions with special emphasis on semantics and reasoning. Developer of SADL.
Varish Mulwad Lead Knowledge Discovery Researcher PhD Computer Science, University of Maryland, Baltimore County	Extraction of domain knowledge and models from documentation and publications	3 years of experience in extracting information and adding semantics to unstructured or semi-structured data by developing and applying techniques from Natural Language Processing, Semantic Web Technologies, Machine Learning, and Probabilistic Graphical Modeling.
Nurali Virani Machine Learning Researcher PhD Mechanical, MS Electrical Pennsylvania State University	K-CHAIN knowledge graph development, inference mechanisms, machine learning algorithms	Expertise in data-driven modeling of physical systems, density estimation, Bayesian learning, Bayesian filtering, optimization theory, graph theory, control theory, sensor fusion, signal processing, and stochastic control. Developing AI-based combustion controllers for gas turbines and performance optimizers for wind turbines.
Narendra Joshi Chief Scientist PhD Mechanical Engineering Stonybrook University	Subject matter expert on science of hypersonic flight	Over 32 years' experience in developing new and advanced products. Developed LM6000 DLE combustion technology, the LMS100 and advanced technologies for application in the GEnx, the GE9x and other product lines.

5. Task Description Document (TDD)

Details of task/subtask objectives, approach, milestones, and deliverables is shown in Table 2.

Table 2: Task Description Document

Task	Objective	Approach	Milestones and Deliverables
Phase 1: Feasibility Study (Base)			
1.1 Establish base domain ontology	Bootstrapping scientific knowledge to enable model-driven extraction from code and text	Identify reusable ontologies, fill gaps with expert-generated	M1: Report initial K-CHAIN, base model content, extraction methodologies M3: Interim report describing prototype system and preliminary capabilities M5: Final report summarizing approach, architecture, data sets, capabilities, and performance
1.2 Extraction from text	Demonstrate feasibility of augmenting the base domain ontology with extraction of concepts, relationships, and processes from domain-specific documentation	Use a custom named entity recognition (NER) system combined with a model-driven approach	
1.3 Extraction from code	Demonstrate feasibility of extracting concepts and relationships from Java code instantiating physics-based models	Look for parsing and analysis tools; buy or build. When parsing isn't possible exercise model and use active learning to characterize	
1.4 Knowledge capture in K-CHAIN	Create knowledge storage capable of representing multiple types of scientific domain knowledge and models	Knowledge-Consistent Hybrid AI Network combining semantic languages, equations, and data-driven models with inference and querying	
1.5 Controlled-English bi-directional interface	Demonstrate human-readable results generated from queries and knowledge exploration	Use and possibly extend Semantic Application Design Language (SADL); enhance translation capabilities from K-CHAIN to natural language	

Task	Objective	Approach	Milestones and Deliverables
Phase 2: Proof of Concept (Option)	Extend capabilities in extraction from code, extraction from text, knowledge representation and reasoning in K-CHAIN, and controlled-English representation for human readability	Based on what is learned in Phase 1, identify gaps, develop solutions, and identify strategies to boost capability and performance in all areas	M6: Milestone Report M7: Report: lessons learned, updated architectures, algorithms, and approaches M9: Report proposed evaluation metrics and initial results of applying metrics measurements to PoC implementation M11: Scripted demonstration of PoC system; report of performance of system on NASA hypersonic aerodynamic models and documents M12: Progress report M13: Interim report quantifying PoC system performance M15: live demonstration of system showing performance metrics for application to NASA hypersonic aerodynamic models and documents; progress report M18: Final report on PoC system architecture, algorithms, and methodology along with performance (quantification of accuracy, robustness) and generalizability

6. Bibliography

[1] A. W. Crapo, *A Cognitive-theoretic Approach to the Visual Representation of Modeling Context,* PhD Thesis: Rensselaer Polytechnic Institute, 2002.

[2] NASA Glenn, "Hypersonic Aerodynamics Index," 13 September 2018. [Online]. Available: https://www.grc.nasa.gov/www/BGH/shorth.html.

[3] "Xtext: Integration with EMF," [Online]. Available: https://www.eclipse.org/Xtext/documentation/308_emf_integration.html. [Accessed 14 Septempber 2018].

[4] T. Aiello, "Safe and Secure Systems and Software Symposium (S5): SoI Requirements Group, Slide 24," 3 August 2017. [Online]. Available: http://www.mys5.org/Proceedings/2017/Day_3/2017-S5-Day3_0905_SoI_Requirements_Group%20_Aiello.pdf. [Accessed 14 September 2018].

[5] H. Yu and B. Meng, "Formal Analysis and Verification of UxAS," Final Report to AFRL, available upon request, Niskayuna, NY, 2017.

[6] I. Augenstein, M. Das, S. Riedel, L. Virraman and A. McCallum, "Task 10: ScienceIE - Extracting Keyphrases and Relations," arXiv:1704.02853, 2017.

[7] G. Y. Kristianto and A. Aizawa, "Extracting textual descriptions of mathematical expressions in scientific papers," *D-Lib Magazine,* 2010.

[8] J. Jin, X. Han and Q. Wang, "Mathematical formulas extraction," in *Seventh International Conference on Document Analysis and Recognition*, 2003.

[9] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami and C. Dyer, "Neural architectures for named entity recognition," *arXiv preprint,* 2016.

[10] Apache, "Apache UIMA ConceptMapper Annotator Documentation," [Online]. Available: https://uima.apache.org/downloads/sandbox/ConceptMapperAnnotatorUserGuide/ConceptMapperAnnotatorUserGuide.html. [Accessed 13 September 2018].

[11] V. Mulwad and K. S. Aggour.US Patent Application 15/836,064: Systems and Methods for Learning to Extract Relations from Text via User Feedback, 2017.

[12] A. Crapo and A. Moitra, "Towards a Unified English-Like Representation of Semantic Models, Data, and Graph Patterns for Subject Matter Experts," *International Journal of Semantic Computing,* vol. 7, no. 3, pp. 215-236, 2013.

[13] A. Crapo, "Semantic Application Design Language (SADL)," [Online]. Available: http://sadl.sourceforge.net/. [Accessed 14 September 2018].

[14] P. Cuddihy, J. McHugh, J. Williams, V. Mulwad and K. Aggour, "SemTK: A Semantics Toolkit for User-friendly SPARQL Generation and Semantic Data Management," in *The 17th International Semantic Web Converence*, Monterey, CA, 2018.

[15] A. Crapo, A. Moitra, C. McMillan and D. Russell, "Requirements Capture and Analysis in ASSERT(TM)," in *2017 IEEE 25th International Requirements Engineering Conference (RE)*, Lisbon, Portugal, 2017.

[16] K. Siu, A. Moitra, A. W. Crapo, H. Chamarthi, M. Durling, M. Li, H. Yu, P. Manolios and M. Meiners, "Towards Development of Complete and Conflict-Free Requirements," in *IEEE Requirements Engineering Conference (RE'18)*, Banff, Canada, 2018.

[17] "Resumes," [Online]. Available: https://drive.google.com/open?id=1mEDojd1D1E4kYNFKxsHvQGoHNIj2Od-q.