# DARPA ASKE TA1 ANSWER Initial Report

## Submitted in accordance with Milestone Phase 1-2.1 under agreement number HR00111990006.

## Introduction

Our focus during the first month of work has been on establishing the base line and identifying the key risk factors and a strategy for their mitigation for each of the 5 tasks of Phase 1. The results of this effort is described for each task in the sections below.

## Task 1.1: Establishment of Base Domain Ontology

We have identified the requirements of our base, domain-independent ontology for ANSWER as well as the extensions necessary to support the NASA Hypersonics domain.

## Task 1.2: Extraction of Scientific Knowledge from Text

We focused on two key aspects (a) extracting scientific equations from text and (b) extracting scientific concepts from text and the relationships between them.

1. Extraction equations: We identified two possible approaches to extract the raw equation from text. The first approach will use a Recurrent Neural Network based architecture to tag equations in a given sentence. The approach will be akin to tagging named entities in a sentence, with the “entity” type in question being an equation. We also identified a less-risk averse heuristic rules-based algorithm to identify the beginning and ending of an equation in a given sentence. After comparing the pros, cons and risks, we selected XXXX approach for the following reasons: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Extracting Scientific Concepts: To identify relations such as dependence, causality etc. between different concepts, the first step would be identifying relevant concepts in a given sentence. Tagging scientific concepts is also similar to tagging named entities in text. We choose to develop a custom named entity recognition (NER) system for this task. In the first month, we completed the task of generating a labelled dataset to train the custom NER. The training dataset was generated by automatically annotating X documents (source of these documents?) with a pre-defined dictionary of scientific concepts. The seed dictionary of concepts was prepared by manually extracting them from sources such as NASA’s Hypersonic Aerodynamics Index, Wikipedia and Wikidata. Apache UIMA’s dictionary based ConceptMapper Annotator was used to tag the training data documents.

## Task 1.3: Extraction of Scientific Knowledge from Code

We have started with the NASA Hypersonics applets, which are coded in Java. After reviewing alternatives, we selected XXXX as the parser for the Java grammar. To maintain as much independence of that selection as possible, we wrap the resulting parse tree in a generic interface that could be used to wrap the output of any parser, for the Java grammar or for other languages. This generic parse tree interface is then exercised to convert the scientific model represented in the code to a semantic model.

## Task 1.4: Representation of Multiple Knowledge Types in K-CHAIN

## Task 1.5: Controlled-English Bi-directional Human-Computer Interface

SADL provides a controlled-English grammar for expressing OWL models plus equations, rules and queries that has proven to be effective for enabling domain experts to understand, capture, and maintain formal models. In prior work, SADL equations permitted only a single output value. However, the need to support probabilistic models and the expectation of interfacing with such models ultimately executed and possibly natively stored in Python suggest a SADL enhancement to support equations returning a list of values. This capability has been implemented in SADL with a small grammar extension.

<example of multi-return value equation in SADL>

Another SADL language extension deemed highly relevant to the task at hand is the introduction of a “where” clause into the equation grammar. The “where” clause is useful in two ways. First, it allows a simplification of representation through modularity. A basic equation can capture in as simple form as possible the underlying physics while intermediate variables can be used and then defined in the “where” clause. For example, …

<example of an equation with a where clause to maintain simplicity, readability>

mdot = (A \* pt/sqrt[Tt]) \* sqrt(gam/R) \* M \* [1 + .5 \* (gam-1) \* M^2 ]^-[(gam + 1)/(gam - 1)/2]

where gam = cp / cv

The other use of “where” in an equation is to capture assumptions. For example,

Force = Mass \* Acceleration

where d(Mass)/d(t) = 0.

While SADL has included an initial prototype of an OWL to SADL import capability, it was not sufficiently robust to support the complexities of the OWL ontologies found to be relevant to ANSWER. Therefore, this import capability has been enhanced to support XXXX.

SADL has focused on the representation of models and the query language supports only a limited subset of SPARQL. Its limitations are less of an issue because queries can be written in native SPARQL, exposing the full power of the query language in SADL. A well-known difficulty with using SPARQL is the complexity and lack of intuitiveness encountered whenever a need arises to query for OWL restrictions and other axiomatic concepts captured in an OWL model. While SADL provides a controlled-English for creating such constructs, there has not, here-to-fore, been a friendly way to retrieve these constructs by query. We have evaluated possible solutions to this need and have also considered other SADL grammar extensions needed to support K-CHAIN probabilistic constructs and to enable query and inference results to be reported in more easily understood representations.

In addition to controlled-English representations, graphical representations are also an effective way of communicating information to human users. We hypothesize that graphical representations are a more effective means of revealing scientific model content and results to users than as a means of knowledge capture from humans. Thus, our focus in the past has been on using graphs to reveal semantic model content. In contemplating the needs of ASKE TA1 (and TA2) we also consider how probabilistic sub-models and how inference and model execution results might be represented graphically.