

Semantic Explorations

1. Log into the [AWS instance](#) and start a session
2. Start Eclipse by clicking on the “eclipse” icon
3. If a Welcome window appears, close it.
4. In the Project Explorer, go to the M5 project and open it if it is not already open
5. Explore the SADL Implicit Model
 - a. Expand the “ImplicitModel” folder and open “SadlImplicitModel.sadl” in a SADL editor
 - b. Note that *UnittedQuantity* is a subclass of *ScientificConcept*. (Not all scientific concepts have units.) Capturing units along with numeric values is an essential part of scientific modeling. This provides a very simple mechanism for doing so and is sufficient for our proof-of-concept. More sophisticated approaches are available.
 - c. Note the *DataDescriptor* class. This captures both simple *dataType* information and semantic information (*augmentedType*) necessary to understand and apply scientific models and to properly use observational data. The *augmentedType* property range is the *AugmentedType* class, which has a number of subclasses that allow representation of a variety of constraints and assumptions about the model.
 - d. An *Equation* has *arguments* which are a list of *DataDescriptor* instances, and *returnTypes*, also a list of *DataDescriptor* instances. A list is used because order of arguments and returned values is essential information.
 - e. A *DataTable* has *columnDescriptors* which is a list of *DataDescriptor* instances.
6. Explore the code extraction meta-model
 - a. Expand the “ExtractedModels” folder and open “CodeExtractionModel.sadl” in a SADL editor
 - b. The top-level class is *CodeElement*, which has a number of subclasses for various elements of the code. All code elements can record a beginning and ending location in the code serialization from which they are extracted.
 - c. *Class* and *Method* are particularly interesting subclasses of *CodeBlock*. In object-oriented programming languages, a class may represents a concept in the domain and a method may represent unit of computation amenable to instantiation as a scientific model in a computational graph. Note that a *CodeBlock* may be *containedIn* another *CodeBlock*, e.g., a *Method* in a *Class*.
 - d. *CodeVariable* is an important concept in code. A *CodeVariable* in Java will have a type (*varType*), e.g., *double*, which will be explicit in its definition or declaration. It may also have a semantic type (*semanticVarType*) identifying what it is in the domain, e.g., *Mass*. A *CodeVariable* will necessarily have more than one *reference*, range *Reference*. It will have one and only one *Reference* with *usage Defined* and may have multiple references with other usages. A *CodeVariable* instance may belong to one of several subclasses depending on where it is defined.
 - e. The rules shown in the current version of the code extraction meta-model reason about when inputs and outputs of a method are implicit. For example, see method *computeMach* in *Mach.java*, which has no explicit inputs or outputs. Class fields are implicit inputs and/or outputs in the class’s methods.
7. Explore basic concepts that might appear in many domains

- a. Open ScientificConcepts2.sadl in a SADL editor
 - b. *Derivative* is a particularly important concept as many scientific concepts are related to others through differentiation. We model *Derivative* in a declarative manner. Examples are given in the definitions of *Velocity*, *Acceleration*, *Momentum*, and *Force*.
 - c. *PhysicalThing* is also an important class and its subclasses *PhysicalObject* and *Substance* capture an important bifurcation of physics.
 - d. The external equation *densityEq* illustrates in a very simple example semantic type information. The input *mass* and *volume* and the returned *density* are all properties of the same *PhysicalThing*.
8. Explore speed of sound from hypersonics
- a. Open SpeedOfSound.sadl in a SADL editor
 - b. Note that it imports the concepts in ScientificConcepts2.sadl
 - c. The contents of this model illustrate the end-goal of extraction from Mach.java and from Sound.txt from the NASA Hypersonics Web site. This content was not extracted by the ANSWER Knowledge Extraction tool, but this is indicative of what a successful knowledge extraction tools would construct. Given the content of the sources, it is clear that a human in-the-loop to assist in completing the scientific knowledge will be essential for some time to come.
 - d. The external equations shown range from quite simple (*specificGasConstantEq*) to quite complex (*computeMach*) illustrate the necessary capture of domain relationships between inputs and outputs and the necessity of property handling units. In the case of *computeMach* (from Mach.java), all of the inputs and outputs are implicit. (The type is void and it has no arguments.)