Extracting of Equations from Text Using the ANSWER Knowledge Extraction Platform

1. Log into the AWS instance and open a new session
2. Start Eclipse by clicking on the "eclipse" icon. (Not the "eclipse-DEV" icon; that runs the plug-in development environment and may be useful for exploring the Dialog and SADL plug-in source code)
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. Open the file "demo.dialog"
   a. Make sure that it is opened in the Dialog editor. If the coloring is not as shown below, close the window, right-click on "demo.dialog" in the Project Explorer, and select Open with -> Dialog Editor

   ```
   1  uri "http://answer/demo.dialog" alias demodlg.
   2
   3  import "http://sadl.org/SpeedOfSound.sadl".
   ```

   b.
6. Uncomment the line "What is Acceleration?" and enter a newline at the end of the statement. The definition of *Acceleration* will be given in structured English. Place the cursor in *Acceleration* (or one of the other model concepts in the statement) and right-click, selecting "Open Declaration" (or press F3). The definition of *Acceleration* (or a different selected concept) will be opened in context, showing the linking of concepts. Alternatively, right-click and select "Find References" to see all of the places where *Acceleration* is used. A Dialog Editor window, with grammar an extension of SADL was chosen in part to facilitate the hyperlinking of concepts to their definitions and usages. Once all CM output of modeling concepts and statements is in the Dialog grammar, all will be linked.
   Note also that content assistance is available in many places by pressing cntrl-space. For example, after "What is " pressing cntrl-space will give a list of known concepts.
7. If you do not do the step above, activate the Dialog Editor window by making some change to demo.dialog, e.g., add a newline or a space somewhere.
8. Right-click on the project M5 in the Project Explorer window, or from the File pulldown menu click on Import… The Import modal dialog will appear.

9. Select Java Model Importer -> Java Files
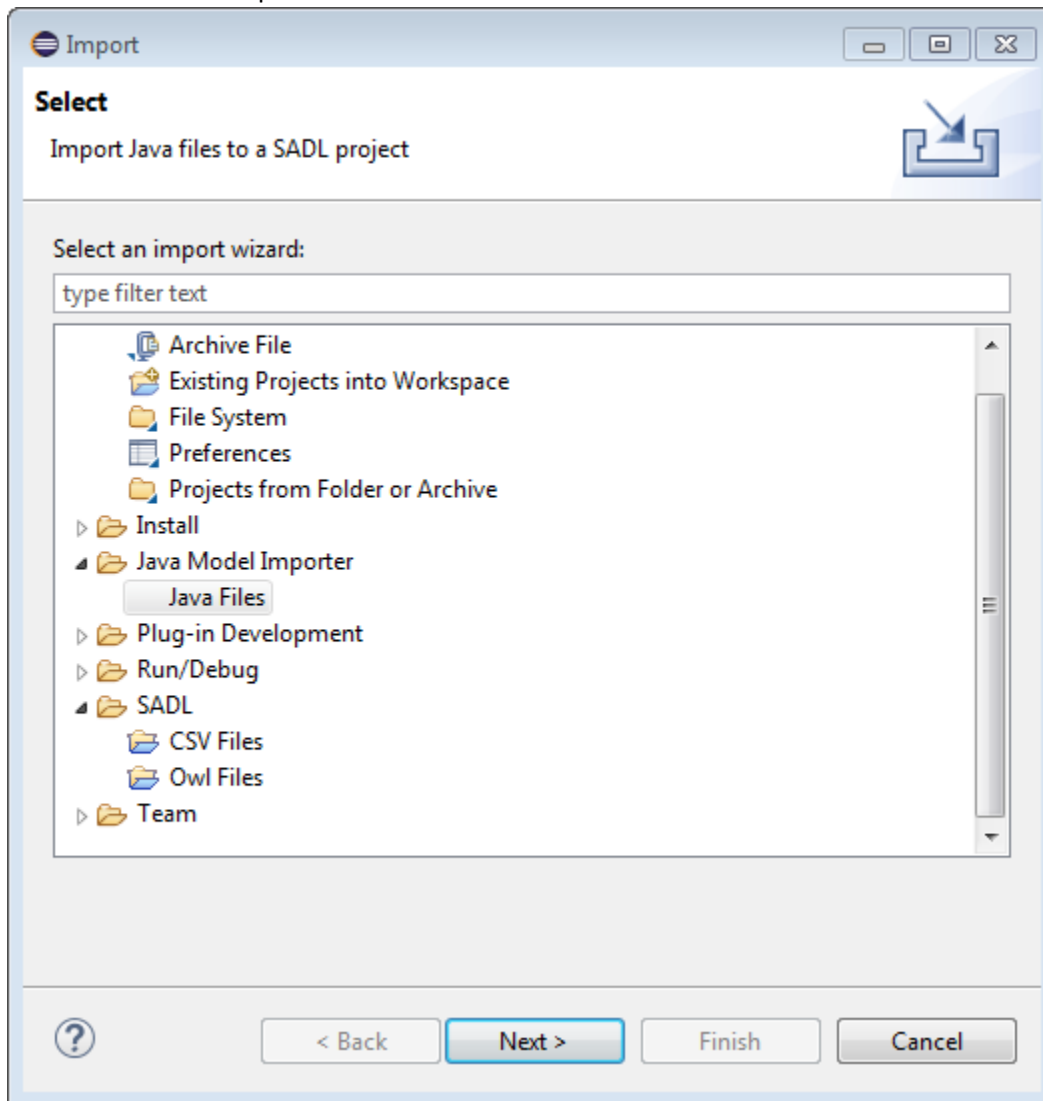
10. Click on Next

11. In the Import Java and Text files modal dialog, first set the "From directory" by browsing to the folder JavaCodeAndTextFiles.
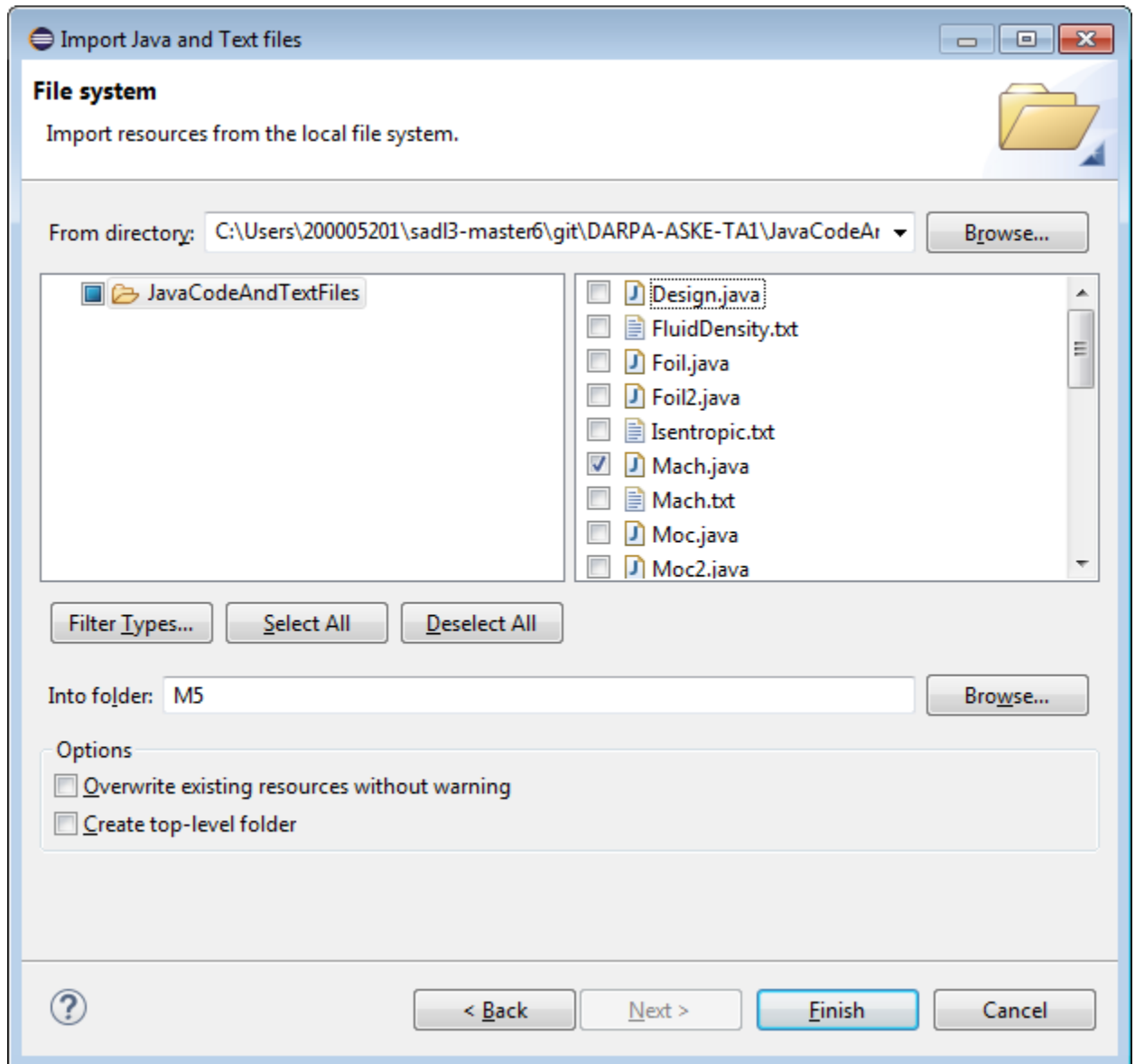
12. Select Java files and/or text (txt) files to import from. A suggested choice, used in the rest of these instructions, are the related files Mach.java and Sound.txt.
13. Select as the "Into folder" the Project, M5.
14. Click on Finish
15. Wait until the modal dialog disappears. The imported files will be opened in editors for examination. Then switch focus back to Dialog Editor window for demo.dialog. The Curation Manager (CM) will notify the user of what happens during the imports and display interesting results. It will then ask the user if the extracted models should be translated to structured English (SADL format).

```
*demo.dialog ⊠   Mach.java      Sound.txt ⊠

15  CM: "Importing text file 'C:/Users/200005201/sadl3-master6/git/DARPA-ASKE-TA1/Ontology/M5/ExtractedModels/Sources/Sound.txt'".
16⊖ CM: "Equations found in extraction:
17  'eq_140577935476560','a^2 = R * T * {1 + (gamma - 1) / ( 1 + (gamma-1) * [(theta/T)^2 * e^(theta/T) /(e^(theta/T) -1)^2]) }'
18  'eq_140577940440960','a^2 = R * T * gamma'
19  ".
20⊖ CM: "To import this model for exploration in this window, add an import at the top of the window (after the 'uri' statement) for URI:
21     http://com.ge.research.sadl.darpa.aske.answer/Sound_txt
22  ".
23  CM: "Importing code file 'C:/Users/200005201/sadl3-master6/git/DARPA-ASKE-TA1/Ontology/M5/ExtractedModels/Sources/Mach.java'".
24⊖ CM: "Interesting methods found in extraction:
25  'Mach.CAL_TT'
26  'Mach.CAL_SOS'
27  'Mach.In.Up.handleBut'
28  'Mach.filter0'
29  'Mach.CAL_T'
30  'Mach.CAL_GAM'
31  'Mach.In.Up.action'
32  'Mach.TQTT'
33  'Mach.In.Up.handleProb'
34  'Mach.filter3'
35  'Mach.CAL_TV'
36  'Mach.In.Titl.In2.handleEvent'
37  'Mach.filter5'
38  ".
39⊖ CM: "To import this model for exploration in this window, add an import at the top of this window (after the 'uri' statement) for URI:
40     http://com.ge.research.sadl.darpa.aske.answer/Mach_java
41  ".
42  CM: "Would you like to save the extracted models (Sound.txt.owl, Mach.java.owl) in SADL format"?
43
```

Note that extraction from text uses the Text-to-Triples REST service. Extraction from code currently shows the methods in Java format but will, once more semantic information is extracted from comments and/or obtained interactively from the user, call the Java-to-Python service to convert the method to Python and then call the K-CHAIN build service to instantiate the model in the computational graph.

16. To see the extracted models as SADL, answer yes ("y." or "yes.") If you answer yes, give the software a moment to accomplish this task. The red marker next to your answer will disappear when the translation is complete. The resulting SADL files will be opened in editors for viewing but focus is automatially moved to these files. Click on an editor tab to see the file. If an error marker appears on the first line of the file, do a project clean/build (Project-> Clean… with Build Automatically checked).

```
*demo.dialog      Mach.java      Sound.txt      Sound.txt.owl.sadl    Mach.java.owl.sadl ⊠

879⊖ Mach.CAL_GAM is a cem:Method,
880⊖     has cem:serialization "public double CAL_GAM(double T, double G, double Q) {
881      return (1 + (G - 1) / (1 + (G - 1) * (Math.pow((Q / T), 2) * Math.exp(Q / T) / Math.pow((Math.exp(Q / T) - 1), 2))));
882⊖ }",
883⊖     has cem:cmReturnTypes ["double"],
884⊖     has cem:cmArguments [Mach.CAL_GAM.T,Mach.CAL_GAM.G,Mach.CAL_GAM.Q],
885      has cem:containedIn Mach.
886⊖ Mach.CAL_GAM.G is a cem:MethodArgument,
887⊖     has cem:reference (a cem:Reference
888⊖     with cem:usage cem:Used,
889⊖     with cem:endsAt 109,
890⊖     with cem:beginsAt 109,
891⊖     with cem:codeBlock Mach.CAL_GAM),
892⊖     has cem:reference (a cem:Reference
893⊖     with cem:usage cem:Used,
894⊖     with cem:endsAt 109,
895⊖     with cem:beginsAt 109,
896⊖     with cem:codeBlock Mach.CAL_GAM),
897⊖     has cem:varName "G",
898⊖     has cem:reference (a cem:Reference
899⊖     with cem:input true,
900⊖     with cem:usage cem:Defined,
901⊖     with cem:endsAt 108,
902⊖     with cem:beginsAt 108,
903      with cem:codeBlock Mach.CAL_GAM).
```

The information extracted from the code provides the necessary basis for reasoning about the code. Once interesting code elements are identified, the original Java code (see lines 889-882 in the figure) will be passed to the Java-to-Python service. Load CodeExtractionModel.sadl into an

SADL editor window to see the code extraction meta-model and some initial rules to help identify implicit inputs and outputs, such as occur throughout the Mach.java code.

```
   *demo.dialog    Mach.java    Sound.txt    Sound.txt.owl.sadl    Mach.java.owl.sadl

 1  uri "http://com.ge.research.sadl.darpa.aske.answer/Sound_txt" alias Sound_txt
 2      (note "This ontology was created by extraction from text by the ANSWER TextProcessor.").
 3
 4
 5
 6  // Individuals:
 7  eq_140464149027048 is a sadlimplicitmodel:ExternalEquation,
 8      has sadlimplicitmodel:expression (a sadlimplicitmodel:Script
 9      with sadlimplicitmodel:language sadlimplicitmodel:Python,
10      with sadlimplicitmodel:script "a = tf.math.pow( R * T * gamma, 1/2)"),
11      has sadlimplicitmodel:expression (a sadlimplicitmodel:Script
12      with sadlimplicitmodel:language sadlimplicitmodel:Text,
13      with sadlimplicitmodel:script "a^2 = R * T * gamma").
14  eq_140464476319944 is a sadlimplicitmodel:ExternalEquation,
15      has sadlimplicitmodel:expression (a sadlimplicitmodel:Script
16      with sadlimplicitmodel:language sadlimplicitmodel:Python,
17      with sadlimplicitmodel:script "a = tf.math.pow( R * T *  (  1 + ( gamma-1 ) / ( 1 + ( gamma-1 ) *  (  ( theta/T )
18      has sadlimplicitmodel:expression (a sadlimplicitmodel:Script
19      with sadlimplicitmodel:language sadlimplicitmodel:Text,
20      with sadlimplicitmodel:script "a^2 = R * T * {1 + (gamma - 1) / ( 1 + (gamma-1) * [(theta/T)^2 * e^(theta/T) /(e^(
```

17. The extracted models are not yet added to the Dialog Editor model imports so at present this is a manual step. To query these models interactively, add the imports URIs as they are indicated by the CM. (Lines 21 and 40 in the Figure of step 14 show where these URIs are given.)

```
   *demo.dialog    Mach.java    Sound.txt    Sound.txt.owl.sadl    Mach.java.owl.sadl

 1  uri "http://answer/demo.dialog" alias demodlg.
 2
 3  import "http://sadl.org/SpeedOfSound.sadl".
 4  import "http://com.ge.research.sadl.darpa.aske.answer/Mach_java".
 5  import "http://com.ge.research.sadl.darpa.aske.answer/Sound_txt".
```

Note that they can also be added as imports using content assist but an unresolved bug results in missing double quotes around the import name which must be supplied manually.

18. A variety of questions can be asked about the extracted models in the Dialog Editor window.

```
Ask select m where m is a Method?
CM: "'m'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.CAL_TT'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.loadInpt'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.CAL_SOS'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.In.Up.handleBut'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.filter0'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.In.insets'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.CAL_T'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.CAL_GAM'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.In.Up.action'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.TQTT'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.In.Up.handleProb'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.init'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.In.Up.insets'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.setDefaults'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.computeMach'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.filter3'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.CAL_TV'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.In.Titl.In2.handleEvent'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.filter5'
'http://com.ge.research.sadl.darpa.aske.answer/Mach_java#Mach.insets'".
```

or the models may be explored in their respective editor windows. Note that the results of

querying, as well as the notifications of interesting results found by the CM during import, are not yet translated to structured English valid in the SADL and Dialog grammars so they are shown for now as quoted strings.

19. Because of the proof-of-concept nature of the software, it may be necessary to exit and restart Eclipse before doing subsequent extractions from code and/or text.