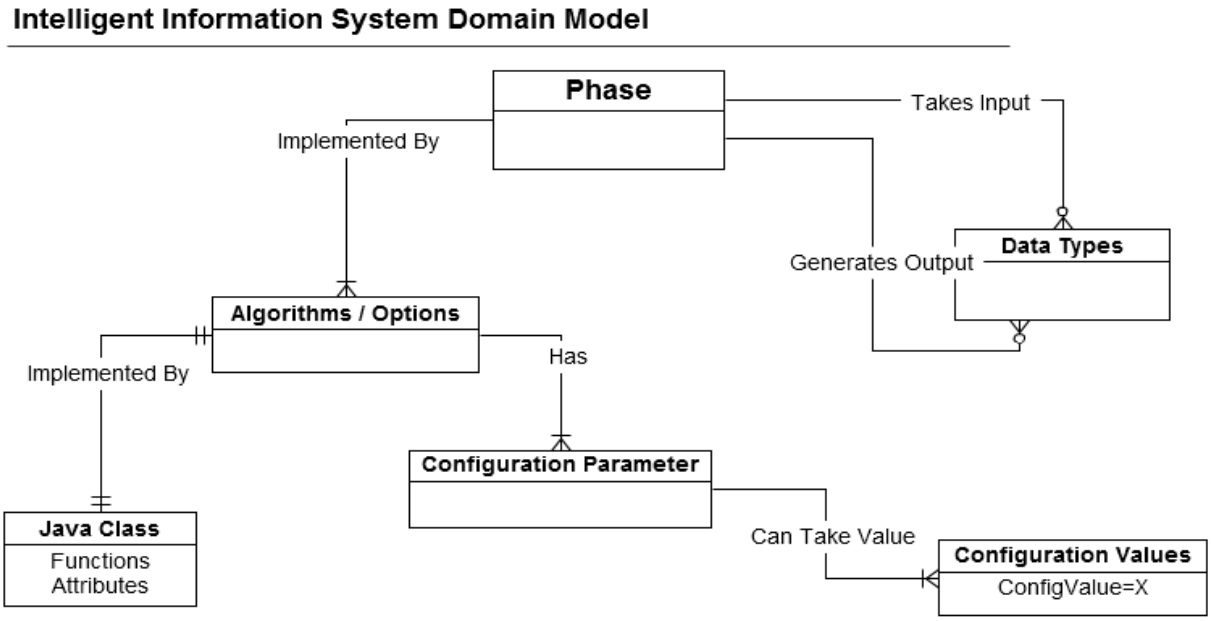
Task1: UML Design

Shourabh Rawat(srawat@cs.cmu.edu)

Task1.1: Domain Diagram for Intelligent Information System

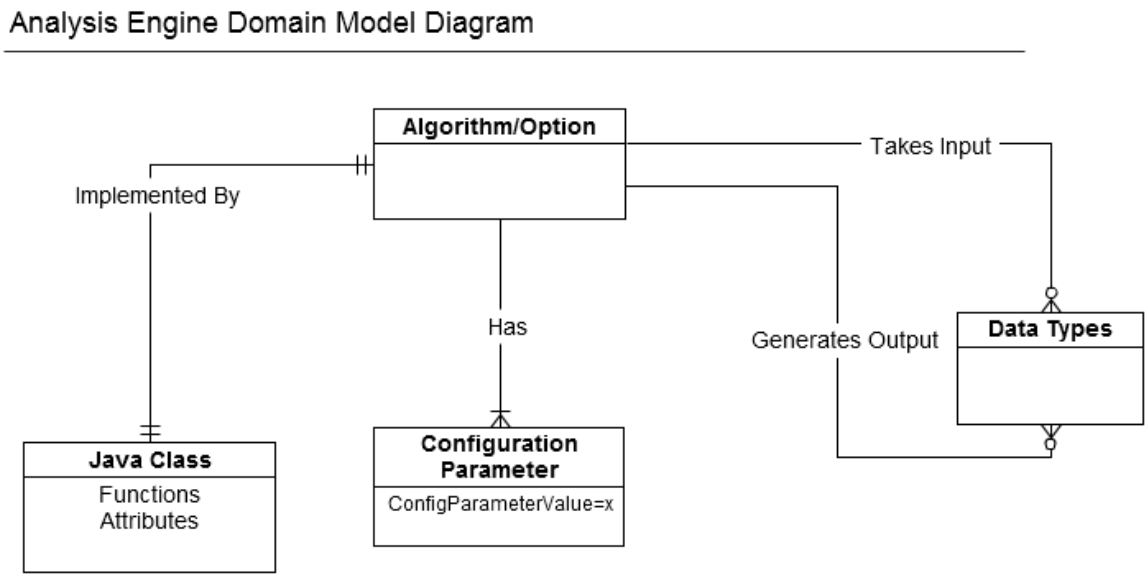


Domain Concepts And Associations with multiplicities:

1. Phase [Represents one of the phase in IIS. Note that we didn't consider IIS as a separate domain concept as it represents the system itself.]
2. Algorithm [Represent Algorithm used in the implementation of a phase. *One or More* ("any number") algorithm can be used to implement a phase.]
3. Java Class [We show a *One to One* mapping between Algorithm and a Java Class]
4. Data Types[They represent the type system which forms the part of the input and the output. We can have *Zero Or More* Inputs and Outputs depending on the phase]
5. Configuration Parameters[Each algorithm can have *One or More* (any number) of configuration parameters]

Since each configuration can take a value from a set of acceptable values, instead of keep configuration value as an attribute to configuration parameter, we took it as a separate domain concept. This is because the "Set of Acceptable Values" is not a primitive data type and too complex to be taken as an attribute. Instead we create it as a separate domain concept showing a *One to Many* "Can Take Value" association with the Configuration Parameter i.e a Configuration Parameter can take one of the many possible values.

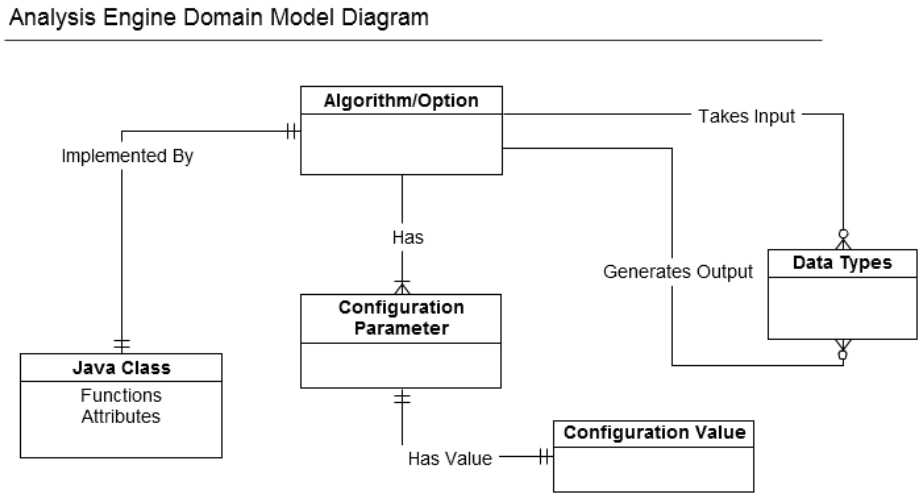
Task1.2: Domain for Analysis Engine



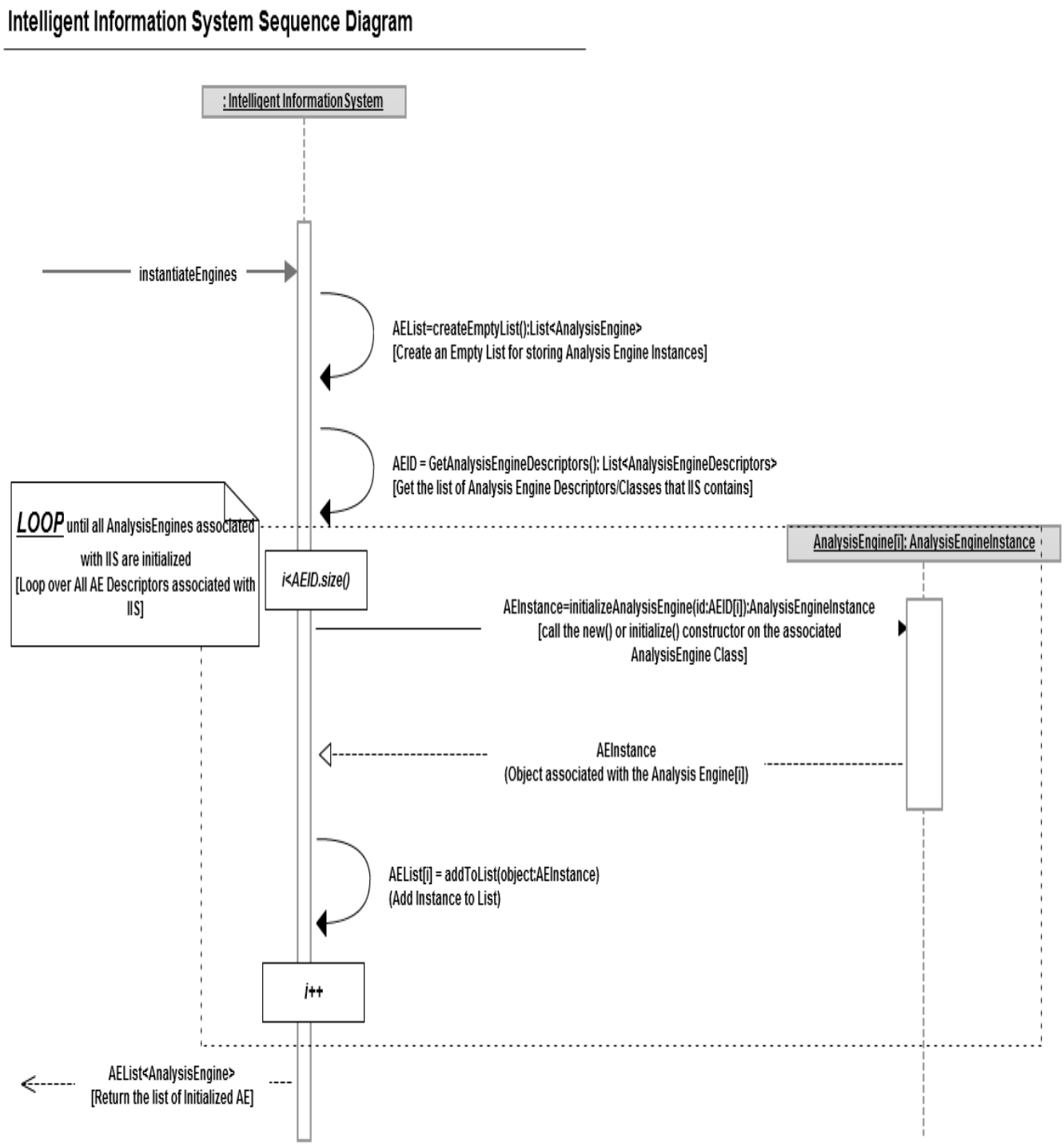
Domain Concepts and Associations with Multiplicity:

1. Algorithm [Represent Algorithm used in the implementation of an Analysis Engine.]
2. Java Class [We show a *One to One* mapping between Algorithm and a Java Class]
3. Data Types[They represent the type system which forms the part of the input and the output. We can have *Zero Or More* Inputs and Outputs depending on the Algorithm]
4. Configuration Parameters[Each algorithm can have Zero or More (any number) of configuration parameters]

Unlike previous case here the Configurations Parameters can only take a single specified value like a string or a primitive data type. Considering this as the case we have considered it as an attribute. If the configuration value is not primitive data type, then again it would make sense to have a separate domain concept for Configuration Value.



Task1.3: Sequence Diagram



If this is not visible clearly, one can look at the better version in the docs folder under the name *"IISSequenceDiagram.png"*