Possible approaches for objects:

1. A generic object with ALL variables where the combination of specific variables defines which kind of object it is
2. A generic object with SOME common variables where specific classes defines which kind of generic object it is

Synonym keywords:
Only use keywords rather than symbols when possible. Decide on ONE keyword for an object. I.e DO NOT use synonyms other than when translating a document when input to the application (ex block will be translated to part def when you input the file).

Connections:
When the parser finds a connection in the form of “connect … to … => ...” it should translate it to a “connection def” and a “connection: ... connect … to …”.

Searching of non-defined objects:
When the parser finds an object that doesn’t have a definition ex “part eng : Engine”, it will search the entire document for a definition, ex “part def Engine”, and then create the definition object first before creating the non-definition object.

Instances:
The array instances() is an array that contains all instances of a type of definition. ex all instances of Engine “part eng : Engine”, “part eng2 : Engine”.

Importing:
The import variable can be used by the GenericClass to keep track of instances.

References:
The reference variable can be used by the GenericClass to keep track of references.

Specialization:
When specializing an object, clone the object definition that is to be specialized and then add any additional features ex. extra parts or attributes. This seems to be the same as “subset”.

Subset:
See specialization.

State transitions and their names:
It is necessary to keep track of the correct indexes of the “to” StateClassArray() and the “transitionNames” StringArray() so that the correct names are paired with the correct “to” states.

Timeslices:
Name is confusing because there is no specification of an amount of time that it lasts. What makes it differ from just an individual?