Report for 2019-03-20 – Collections and Patterns

## Patterns

* Should there be a separate level of model that contains the patterns and then a level derived from that that removes the patterns for users (automatedly implementing the pattern in “realized”/ inherited classes)
* Should we use inheritance to describe the imposition of the patterns?
	+ This could imply the use of multiple-inheritance for a few cases. But could be removed in the lower level of model described above. Super-classes are not allowed to inherit from the same class (removing Identifiable would fix this in most if not all cases)

Look at analytics on associations in the model – for Packages and Views

 Cluster analysis

 Class to class

 Package to package and View to View

 Look at modeling associations in network analysis tool (e.g. Gephi)

Most of the multiple inheritance in the model is due to Identifiable and Annotated Identifiable

 We can model this differently – e.g. Identifiable just becomes a property with a set of properties to be repeated

Pull associations out of structured datatype e.g. particularly Annotation

 Many currently have associations. These should be classes or remodeled

Other Structured datatypes (without associations) should be refactored as UML datatypes

Express Views as an aggregation of classes (see below).

## Requirements for a Collection

* Group a set of entities together
* Describe a structure on that group (or more than one structure)
	+ Simple list (for convenience) not functionally different from complex relationship
		- Allow all four combinations of ordered and unique
	+ Complex relationships (e.g. described by an adjacency list)
		- Allow for at least the types currently listed in RelationSpecification
* Describe multiple structures on the group?
* Constrain membership to particular classes
* Allow ad-hoc set of classes for membership? (e.g. to describe a particular set of metadata)

We looked at alternate ways of modeling the Collections pattern, but did not reach a conclusion

Further discussion of Collections is next. Also review StructuredDatatypes.

