Hilde questions addendum

***1) What are the weaknesses or flaws detected in the current version of the CollectionsPattern (as currently in Lion)?***

While the application of patterns is not integral to a CollectionsPattern or any other pattern, the complexity of a pattern has huge consequences for the model as a whole because, to the extent that it is used and to the extent that it is complicated, it will grow both the number of classes and relationships in the model.

Because the CollectionsPattern in the prototype was BIG and because of the modeling mechanism that was used to apply it and because there are as many as 15 or more applications of the CollectionsPattern, we wanted to reduce its complexity, come up with a mechanism for applying it that was UML compliant and limit the number of applications.

With respect to its complexity this has been somewhat reduced by not using structured datatypes which simply hid lots of complexity and makes collections harder to understand.

We are still weighing a couple of mechanisms for applying design patterns. Both will reduce the number of concrete classes in the model. One approach uses UML abstraction and the other uses UML [interface realization](https://www.uml-diagrams.org/realization.html). We are leaning towards interface realization because interfaces don’t need to count as classes in a UML model. Note that interface realization is a specialized abstraction relationship. In the prototype concrete classes were employed to indicate the pattern in which a collection participates.

Finally, we are coming up with modeling principles for when to use collections and when to use UML aggregation This will impact several packages that exploded with new classes because of collections. Take Representations, for example. Do ClassificationFamily and ClassificationSeries have to be represented as collections in addition to the main attraction, i.e. StatisticalClassification?

***4. Does the proposed changes affect the way other types of collections than instance variable groups are handled in the model, for example concept systems and code lists, and if so how ?***

Modeling principles on when to use UML aggregation vs collections will impact most packages and hopefully reduce both the size and complexity of the entire model.

Also, within the CollectionsPattern and more generally throughout the model we are developing modeling principles for associations. The idea is to make almost all associations navigable just in one direction. Exceptions would need substantive domain-driven rationales. There are many consequences here for the model which we are in the process of documenting.