EDIT - bug #8496

Improve public setters for entity collections

08/28/2019 02:20 PM - Andreas Müller

Status:	New	**Start date:**	
Priority:	New	**Due date:**	
Assignee:	Andreas Kohlbecker	**% Done:**	0%
Category:	cdmlib	**Estimated time:**	0:00 hour
Target version:	Unassigned CDM tickets		
Severity:	normal	**Found in Version:**	

Description

These were implemented in #7600.

Unfortunately the current semantics is that e.g. AnnotatableEntity.setAnnotations(set) does not replace the existing set by the new one but only copies the annotations to the existing set.
This is not what one expects if using a setter so it breaks the semantics. Rather it is a merge into then a set semantics and is something one would expect by an addAll method but not by a set method.

Related issues:

Copied from EDIT - feature request #7600: public setters for entity collections	**Closed**

History

#1 - 08/28/2019 02:20 PM - Andreas Müller

- Copied from feature request #7600: public setters for entity collections added

#2 - 08/28/2019 02:22 PM - Andreas Müller

- Description updated

#3 - 08/28/2019 02:35 PM - Andreas Kohlbecker

- Status changed from New to Feedback

- Assignee changed from Andreas Kohlbecker to Andreas Müller

Does this implementation cause any real problems or is it only semantically problem?

As far as I remember the culprit here is that business logic often exists in the add* and remove* methods. Therefore a semantically classical setter is not possible bu a setter method is still needed anyway. What do you suggest?

#4 - 08/28/2019 03:09 PM - Andreas Müller

- Status changed from Feedback to New

- Assignee changed from Andreas Müller to Andreas Kohlbecker

I had a problem when writing the CDM2CDM import. Here I wanted to call the setter and replace the old collection (which I can't use because it belongs to an other hibernation session) with the new collection. But this was not possible via the setter as the set was not really replaced.

One solution could be to first copy the data to a temporary set, then emptying the new set and replace the old set by the new set and then run the add method from the temporary set (and fill the new set this way).
There might be other solutions, too.

Note: I assigned this to you only because you had the old ticket and you know the requirments for the vaddin usecase. Doesn't mean necessarily that you have to fix it.