

TYPO3.Flow - Bug # 29416

Status:	Resolved	Priority:	Must have
Author:	Karsten Dambekalns	Category:	Persistence
Created:	2011-09-01	Assigned To:	Karsten Dambekalns
Updated:	2011-09-30	Due date:	
PHP Version:			
Has patch:			
Complexity:			
Affected Flow version:			
Subject:	Aggregate root detection for Person vs AbstractParty is broken		
Description			
The association from Account to Person is cascaded during persist, because the fact that Person has a PersonRepository is not detected from the type hint to AbstractParty in the Account class.			
Related issues:			
related to TYPO3.Flow - Task # 29543: Clarify persistence behavior for inher...		Resolved	2011-09-07

Associated revisions

Revision 23961c89 - 2011-09-30 12:20 - Karsten Dambekalns

[!!!][BUGFIX] Make sure only complete hierarchies are aggregate root

Reported as "Aggregate root detection for Person vs AbstractParty is broken" the issue at hand was that one could not rely on certain semantics bound to aggregate root handling:

The association from Account to Person is cascaded during persist, because the fact that Person has a PersonRepository is not detected from the type hint to AbstractParty in the Account class.

The solution this change introduces is as follows: all classes in a hierarchy must have a repository assigned. This can be done by having one repository on the tip of the hierarchy, but any entity can have it's own repository as well.

Change-Id: I5538230b42624629a6746ebff5e9e5dae6e93859

Fixes: #29416

History

#1 - 2011-09-07 17:56 - Karsten Dambekalns

- Status changed from Accepted to Needs Feedback

What actually happens

1. Account has an association to AbstractParty
2. the metadata mapping is done statically and does not find AbstractParty to be an aggregate root
3. cascade={"all"} is added by default
4. during runtime the associated party is actually a Person, which is an aggregate root
5. changes are (falsely) persisted, because of the metadata mapping

What should happen

Depending on what actually is the associated party, operations should cascade or not

The problem

The metadata mapping is static, Doctrine does no runtime type inspection or something. Thus we cannot "simply add a runtime check in the persistence code" or something like that. Doctrine expects the information in the mapping to be "final and correct". If an association to some hierarchy is configured, Doctrine would expect all instances to behave the same way, i.e. cascade options apply to all instances the same way.

The solution?

"Easy"

We can "solve" this problem by making the same assumption: If AbstractParty is an aggregate root, then all concrete subclasses are aggregate roots - if not, none are.

"Complex as hell"

Find a place to hook in some runtime checking code. That code would require persist operations to be configured non-cascading and would then add persist/remove/add calls as needed if an actual associated instance would need it.

#2 - 2011-09-08 12:33 - Christopher Hlubek

I think it should depend on the actual type declaration of a property (the "easy" solution). So for example we could have an AbstractParty that is not an aggregate root and a property declaration with Person, which would be an aggregate root:

```
AbstractParty -> Person
```

```
PersonRepository
```

```
...
```

```
MyAccount:
```

```
party: Person
```

In this case the party would not be persisted (since it's an aggregate root).

We should forbid this declaration, since you could mix aggregate roots and non aggregate roots for the party:

```
AbstractParty -> Person
```

```
PersonRepository
```

```
...
```

```
OtherAccount:
```

```
party: AbstractParty
```

But I would not infer the aggregate root information from the start of the type hierarchy, which would circumvent modeling of your domain objects. It should be rather that once a class is an aggregate root, all subclasses should also be aggregate roots.

#3 - 2011-09-09 15:17 - Robert Lemke

- Target version changed from 1.0 beta 2 to 1.0.0

#4 - 2011-09-14 12:39 - Karsten Dambekalns

- *Status changed from Needs Feedback to Accepted*

Here is what will happen:

- hierarchies need to be consistent - either all members are aggregate root, or none
- if violated (as it is currently the case for AbstractParty and Person), an exception will be thrown

This way we can safely rely on what the "tip" of the hierarchy is and the problem is solved.

#5 - 2011-09-29 23:21 - Mr. Hudson

- *Status changed from Accepted to Under Review*

Patch set 1 of change I5538230b42624629a6746ebff5e9e5dae6e93859 has been pushed to the review server.

It is available at <http://review.typo3.org/5450>

#6 - 2011-09-30 12:20 - Mr. Hudson

Patch set 2 of change I5538230b42624629a6746ebff5e9e5dae6e93859 has been pushed to the review server.

It is available at <http://review.typo3.org/5450>

#7 - 2011-09-30 12:24 - Karsten Dambekalns

- *Status changed from Under Review to Resolved*

- *% Done changed from 0 to 100*