TYPO3.Flow - Task # 29543

Status:	Resolved	**Priority:**	Should have
Author:	Karsten Dambekalns	**Category:**	Persistence
Created:	2011-09-07	**Assigned To:**	Karsten Dambekalns
Updated:	2011-09-30	**Due date:**	
Sprint:			
PHP Version:			
Has patch:			
Complexity:			
Subject:	Clarifiy persistence behavior for inheritance in entities / aggregate roots		

Description

For aggregate roots being part of an inheritance chain, how should their repositories behave? Consider this example:

```
  AbstractParty
    +- Person
       +- User
```

A PersonRepository exists, no UserRepository exists.

1. Should the PersonRepository also handle User instances?
2. Should it be possible to switch such behavior on/off?
3. Or do we require developers to create a repository for every entity in such a chain?
4. are the above options to be combined?

Related issues:

related to TYPO3.Flow - Bug # 29416: Aggregate root detection for Person vs A...	**Resolved**	**2011-09-01**

Associated revisions

Revision 095a742b - 2011-09-30 12:20 - Karsten Dambekalns

[TASK] Clarify persistence behavior for inheritance in aggregate roots

This adds a number of functional tests to make sure the repository
behaves as expected. For Person and User, the latter being a subclass
of the former, a PersonRepository will also be able to handle User:

- it will return both types
- it will accept both types for add/remove/update

Change-Id: Ic64e8cc7b72fc1f570ab4ed9d09a435707e0cc12
Resolves: #29543

History

#1 - 2011-09-07 18:03 - Karsten Dambekalns

- Status changed from Accepted to Needs Feedback

#2 - 2011-09-08 12:25 - Christopher Hlubek

For me it's necessary that subclasses are handled by a repository. So the PersonRepository would also handle an User. That would also be in sync with OOP concepts. If I specify a type Person, I can also pass a more special Person: an User.

Otherwise one had to create a lot of repository bloat for complex type hierarchies.

#3 - 2011-09-08 12:34 - Aske Ertmann

I agree with Christopher. I would add that I would expect that if there is a UserRepository present, it would handle the User. Don't know if the configuration to switch the behavior off is necessary.

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

- Target version changed from 1230 to 1.0.0

As in the given example (Person and User), a PersonRepository will also be able to handle User

- it will return both types
- it will accept both types for add/remove/update
- if more specialized repositories are available, hand over to them

- handling of sub-types can be switched off (on by default)

#5 - 2011-09-29 12:38 - Karsten Dambekalns

- Status changed from Needs Feedback to Accepted

#6 - 2011-09-30 11:52 - Karsten Dambekalns

A small update to the scope of this:

Karsten Dambekalns wrote:

> - *it will return both types*
> - *it will accept both types for add/remove/update*
> - ~~*if more specialized repositories are available, hand over to them*~~
>
> - ~~*handling of sub-types can be switched off (on by default)*~~

#7 - 2011-09-30 12:00 - Mr. Hudson

- Status changed from Accepted to Under Review

Patch set 1 of change Ic64e8cc7b72fc1f570ab4ed9d09a435707e0cc12 has been pushed to the review server.
It is available at http://review.typo3.org/5455

#8 - 2011-09-30 12:20 - Mr. Hudson

Patch set 2 of change Ic64e8cc7b72fc1f570ab4ed9d09a435707e0cc12 has been pushed to the review server.
It is available at http://review.typo3.org/5455

#9 - 2011-09-30 12:25 - Karsten Dambekalns

- Status changed from Under Review to Resolved

- % Done changed from 0 to 100