

Unlinked Roles

Joop Ringelberg

05-01-21

Version: 1

Introduction

Role- and context instances are represented in the PDR by mutually referring data structures. A reference consists of an identifier that we use to retrieve the data structure from the database.

For a non-functional role, a context holds an array of references: one for each of its instances. The number of instances may grow very large for some types of roles. This may make the memory cost of caching their context prohibitive.

Retrieve with a query instead of by identifier

We offer the modeller a ‘compiler instruction’ to use with a role definition: the keyword `unlinked`. It can be used as follows:

```
context: Chats (not mandatory, not functional, unlinked) filledBy: Chat
```

(taken from `model:SimpleChat`).

The role `Chats` will not figure in instances of its context, `ChatApp`. Instead, when query evaluation proceeds with the step `Chats`, as in:

```
user: Chatter (mandatory, functional) filledBy: sys:PerspectivesSystem$User
  perspective on: Chats >> binding >> context >> Initiator: Create,
  Bind, Change
```

it will perform a query on the database for all role instances of the type `model:SimpleChat$ChatApp$Chats`, whose context is the context the query tries to get the `Chats` from.

Semantically, there is no difference between linked roles (the default) and unlinked roles.

Reversing over an unlinked role

Interestingly, we outfit the role instances with a direct reference to their context - just like instances of linked roles. This means that when the query evaluator encounters a `context` step, it handles both cases in the same way.

Deleting unlinked role instances

On deleting an unlinked role instance, we remove it from the database, just like with an instance of a linked role. However, there is no need to remove its reference from its context.