UnitOfWorkProtocol

Overview

The **UnitOfWorkProtocol** aspect makes the Unit Of Work management transparent in the user code.

It will allow you to write your code like this:

```
AddressBook book = new AddressBook(...);
book.addContact(contact);
```

Instead of like this:

```
UnitOrWork unitOfWork = UnitOfWork.begin();
try {
    AddressBook book = new AddressBook(...);
    unitOfWork.registerNew(book);
    book.addContact(contact);
    unitOfWork.registerDirty(book);
    unitOfWork.commit();
}
catch(Exception e) {
    unitOfWork.rollback();
}
finally {
    unitOfWork.dispose();
}
```

and still all the transaction and persistence management will take place.

Definition

The **UnitOfWorkProtocol** aspect has abstract pointcuts that you have to define in your aop.xml file (a "transactional object" is an object that is set to participate in the Unit Of Work transaction, meaning that the Unit Of Work will keep track of its state):

- `txRequires` - picks out all points in the code where you want a *TX REQUIRES* transaction to *begin*, *commit* and *rollback*
- `txRequiresNew` - picks out all points in the code where you want a *TX REQUIRES_NEW* transaction to *begin*, *commit* and *rollback*
- `txSupports` - picks out all points in the code where you want a *TX SUPPORTS* transaction to *begin*, *commit* and *rollback*
- `txMandatory` - picks out all points in the code where you want a *TX MANDATORY* transaction to *begin*, *commit* and *rollback*
- `txNever` - picks out all points in the code where you want a *TX NEVER* transaction to *begin*, *commit* and *rollback*
- `transactionalObjects` - picks out all transactional object, this pointcut is used by the Transactional Mixin
- `transactionalObjectCreationPoints` - picks out all points in the code where a transactional object is created
- `transactionalObjectModificationPoints` - picks out all points in the code where a transactional object is being modified (best done with a `set(...)` pointcut)

Here is an example on how to define this in XML:

```
<aspect
class="org.codehaus.aware.unitofwork.UnitOfW
orkProtocol">
    <pointcut name="txSupports"
expression="execution(*
org.codehaus.aware.app.service.*.*(..))"/>
    <pointcut name="txRequires"
expression="execution(*
org.codehaus.aware.app.domain.*.*(..))"/>

    <pointcut name="transactionalObjects"
expression="within(org.codehaus.aware.app.do
main.*)"/>
    <pointcut
name="transactionalObjectCreationPoints"
expression="call(org.codehaus.aware.app.doma
in.*.new(..))"/>
    <pointcut
name="transactionalObjectModificationPoints"
expression="set(*
org.codehaus.aware.app.domain.*.*)"/>
</aspect>
```