Redmine - Defect #18237

From a rake task context, impossible to create an IssueRelation normally

2014-10-30 04:04 - Stephane Lapie

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Jean-Philippe Lang	**% Done:**	0%
Category:	Code cleanup/refactoring	**Estimated time:**	0.00 hour
Target version:	3.0.0		
Resolution:	Fixed	**Affected version:**	

Description

I am currently doing a rewrite of the Scarab migration script for my company (based mostly on #2928), and have been having a lot of luck recreating the ticket hierarchy.

However, in our case, our Scarab tracker had URL attachments referencing Redmine, so I had a user request to convert that to "#<MigratedScaraTicketb> relates to #<PreviousRedmineTicket>".

This looked easy enough to implement on paper :

```
relation = IssueRelation.new :relation_type => IssueRelation::TYPE_RELATES,
                             :issue_from => Issue.find(id),
                             :issue_to => related_issue
unless r.save
  p r.errors # if $DBG
  next
end
```

I tried running the above code from the "rails console" command line, and it worked just fine.

However, this code blows up in my face upon task execution via rake with the following exception :

```
PG::NotNullViolation: ERROR:  null value in column "created_on" violates not-null constraint
: INSERT INTO "journals" ("created_on", "journalized_id", "journalized_type", "notes", "private_no
tes", "user_id") VALUES ($1, $2, $3, $4, $5, $6) RETURNING "id"
```

and the following stack trace :

```
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/connection_adapters/postgresql_adap
ter.rb:1176:in `get_last_result'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/connection_adapters/postgresql_adap
ter.rb:1176:in `exec_cache'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/connection_adapters/postgresql_adap
ter.rb:661:in `block in exec_query'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/connection_adapters/abstract_adapte
r.rb:280:in `block in log'
/var/lib/gems/1.9.1/gems/activesupport-3.2.19/lib/active_support/notifications/instrumenter.rb:20:
in `instrument'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/connection_adapters/abstract_adapte
r.rb:275:in `log'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/connection_adapters/postgresql_adap
ter.rb:659:in `exec_query'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/connection_adapters/abstract/databa
se_statements.rb:63:in `exec_insert'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/connection_adapters/abstract/databa
se_statements.rb:90:in `insert'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/connection_adapters/abstract/query_
cache.rb:14:in `insert'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/relation.rb:66:in `insert'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/persistence.rb:367:in `create'
```

```
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/timestamp.rb:58:in `create'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/callbacks.rb:268:in `block in creat
e'
/var/lib/gems/1.9.1/gems/activesupport-3.2.19/lib/active_support/callbacks.rb:414:in `_run__176944
9425508713363__create__622663407774382100__callbacks'
/var/lib/gems/1.9.1/gems/activesupport-3.2.19/lib/active_support/callbacks.rb:405:in `__run_callba
ck'
/var/lib/gems/1.9.1/gems/activesupport-3.2.19/lib/active_support/callbacks.rb:385:in `_run_create_
callbacks'
/var/lib/gems/1.9.1/gems/activesupport-3.2.19/lib/active_support/callbacks.rb:81:in `run_callbacks
'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/callbacks.rb:268:in `create'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/persistence.rb:348:in `create_or_up
date'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/callbacks.rb:264:in `block in creat
e_or_update'
/var/lib/gems/1.9.1/gems/activesupport-3.2.19/lib/active_support/callbacks.rb:447:in `_run__176944
9425508713363__save__622663407774382100__callbacks'
/var/lib/gems/1.9.1/gems/activesupport-3.2.19/lib/active_support/callbacks.rb:405:in `__run_callba
ck'
/var/lib/gems/1.9.1/gems/activesupport-3.2.19/lib/active_support/callbacks.rb:385:in `_run_save_ca
llbacks'
/var/lib/gems/1.9.1/gems/activesupport-3.2.19/lib/active_support/callbacks.rb:81:in `run_callbacks
'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/callbacks.rb:264:in `create_or_upda
te'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/persistence.rb:84:in `save'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/validations.rb:50:in `save'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/attribute_methods/dirty.rb:22:in `s
ave'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/transactions.rb:259:in `block (2 le
vels) in save'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/transactions.rb:313:in `block in wi
th_transaction_returning_status'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/connection_adapters/abstract/databa
se_statements.rb:192:in `transaction'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/transactions.rb:208:in `transaction
'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/transactions.rb:311:in `with_transa
ction_returning_status'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/transactions.rb:259:in `block in sa
ve'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/transactions.rb:270:in `rollback_ac
tive_record_state!'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/transactions.rb:258:in `save'
/usr/local/share/redmine-2.5/app/models/journal.rb:54:in `save'
/usr/local/share/redmine-2.5/app/models/issue_relation.rb:190:in `create_journal_after_create'
/var/lib/gems/1.9.1/gems/activesupport-3.2.19/lib/active_support/callbacks.rb:405:in `_run__146320
5190551789637__create__622663407774382100__callbacks'
/var/lib/gems/1.9.1/gems/activesupport-3.2.19/lib/active_support/callbacks.rb:405:in `__run_callba
ck'
/var/lib/gems/1.9.1/gems/activesupport-3.2.19/lib/active_support/callbacks.rb:385:in `_run_create_
callbacks'
/var/lib/gems/1.9.1/gems/activesupport-3.2.19/lib/active_support/callbacks.rb:81:in `run_callbacks
'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/callbacks.rb:268:in `create'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/persistence.rb:348:in `create_or_up
date'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/callbacks.rb:264:in `block in creat
e_or_update'
/var/lib/gems/1.9.1/gems/activesupport-3.2.19/lib/active_support/callbacks.rb:436:in `_run__146320
5190551789637__save__622663407774382100__callbacks'
/var/lib/gems/1.9.1/gems/activesupport-3.2.19/lib/active_support/callbacks.rb:405:in `__run_callba
ck'
/var/lib/gems/1.9.1/gems/activesupport-3.2.19/lib/active_support/callbacks.rb:385:in `_run_save_ca
llbacks'
/var/lib/gems/1.9.1/gems/activesupport-3.2.19/lib/active_support/callbacks.rb:81:in `run_callbacks
```

```
'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/callbacks.rb:264:in `create_or_upda
te'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/persistence.rb:84:in `save'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/validations.rb:50:in `save'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/attribute_methods/dirty.rb:22:in `s
ave'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/transactions.rb:259:in `block (2 le
vels) in save'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/transactions.rb:313:in `block in wi
th_transaction_returning_status'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/connection_adapters/abstract/databa
se_statements.rb:192:in `transaction'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/transactions.rb:208:in `transaction
'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/transactions.rb:311:in `with_transa
ction_returning_status'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/transactions.rb:259:in `block in sa
ve'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/transactions.rb:270:in `rollback_ac
tive_record_state!'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/transactions.rb:258:in `save'
/usr/local/share/redmine-2.5/lib/tasks/migrate_from_scarab.rake:994:in `block (2 levels) in migrat
e'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/associations/collection_proxy.rb:91
:in `each'
/var/lib/gems/1.9.1/gems/activerecord-3.2.19/lib/active_record/associations/collection_proxy.rb:91
:in `method_missing'
/usr/local/share/redmine-2.5/lib/tasks/migrate_from_scarab.rake:949:in `block in migrate'
/usr/local/share/redmine-2.5/lib/tasks/migrate_from_scarab.rake:748:in `each'
/usr/local/share/redmine-2.5/lib/tasks/migrate_from_scarab.rake:748:in `migrate'
/usr/local/share/redmine-2.5/lib/tasks/migrate_from_scarab.rake:1150:in `block (2 levels) in <top
(required)>'
/var/lib/gems/1.9.1/gems/rake-10.1.1/lib/rake/task.rb:236:in `call'
/var/lib/gems/1.9.1/gems/rake-10.1.1/lib/rake/task.rb:236:in `block in execute'
/var/lib/gems/1.9.1/gems/rake-10.1.1/lib/rake/task.rb:231:in `each'
/var/lib/gems/1.9.1/gems/rake-10.1.1/lib/rake/task.rb:231:in `execute'
/var/lib/gems/1.9.1/gems/rake-10.1.1/lib/rake/task.rb:175:in `block in invoke_with_call_chain'
/usr/lib/ruby/1.9.1/monitor.rb:211:in `mon_synchronize'
/var/lib/gems/1.9.1/gems/rake-10.1.1/lib/rake/task.rb:168:in `invoke_with_call_chain'
/var/lib/gems/1.9.1/gems/rake-10.1.1/lib/rake/task.rb:161:in `invoke'
/var/lib/gems/1.9.1/gems/rake-10.1.1/lib/rake/application.rb:149:in `invoke_task'
/var/lib/gems/1.9.1/gems/rake-10.1.1/lib/rake/application.rb:106:in `block (2 levels) in top_level
'
/var/lib/gems/1.9.1/gems/rake-10.1.1/lib/rake/application.rb:106:in `each'
/var/lib/gems/1.9.1/gems/rake-10.1.1/lib/rake/application.rb:106:in `block in top_level'
/var/lib/gems/1.9.1/gems/rake-10.1.1/lib/rake/application.rb:115:in `run_with_threads'
/var/lib/gems/1.9.1/gems/rake-10.1.1/lib/rake/application.rb:100:in `top_level'
/var/lib/gems/1.9.1/gems/rake-10.1.1/lib/rake/application.rb:78:in `block in run'
/var/lib/gems/1.9.1/gems/rake-10.1.1/lib/rake/application.rb:165:in `standard_exception_handling'
/var/lib/gems/1.9.1/gems/rake-10.1.1/lib/rake/application.rb:75:in `run'
/var/lib/gems/1.9.1/gems/rake-10.1.1/bin/rake:33:in `<top (required)>'
```

I quickly could trace the source of the problem, back to app/models/issue_relations.rb and app/models/issue.rb :

- app/models/issue_relations.rb :

```
def create_journal_after_create
  journal = issue_from.init_journal(User.current)
  journal.details << JournalDetail.new(:property => 'relation',
                                       :prop_key => relation_type_for(issue_from),
                                       :value    => issue_to.id)
  journal.save
  journal = issue_to.init_journal(User.current)
  journal.details << JournalDetail.new(:property => 'relation',
                                       :prop_key => relation_type_for(issue_to),
                                       :value    => issue_from.id)
```

```
    journal.save
  end
```

- app/models/issue.rb :

```
  def init_journal(user, notes = "")
    @current_journal ||= Journal.new(:journalized => self, :user => user, :notes => notes)
    if new_record?
      @current_journal.notify = false
    else
      @attributes_before_change = attributes.dup
      @custom_values_before_change = {}
      self.custom_field_values.each {|c| @custom_values_before_change.store c.custom_field_id, c
.value }
    end
    @current_journal
  end
```

Basically, the code will create on its own a journal entry to take notes about the relation that was just created, transparently. However, I am running this from a rake task and not a web/REST access. This means :

- issue_from.init_journal will rely on whatever context it has at hand (current_journal, @User.current)
- The caller code has absolutely no control over the generated journal entry. (who owns it, when it is marked as created, etc...)

In the end, I patched app/models/issue_relations.rb for my specific case, for the time of my migration :

```
--- /tmp/issue_relation.rb    2014-10-30 10:44:07.719760399 +0900
+++ /usr/local/share/redmine/app/models/issue_relation.rb    2014-10-30 03:07:46.461474559 +0900
@@ -184,11 +184,13 @@

   def create_journal_after_create
     journal = issue_from.init_journal(User.current)
+    journal.created_on = issue_from.created_on if journal.created_on.nil?
     journal.details << JournalDetail.new(:property => 'relation',
                                          :prop_key => relation_type_for(issue_from),
                                          :value    => issue_to.id)
     journal.save
     journal = issue_to.init_journal(User.current)
+    journal.created_on = issue_to.created_on if journal.created_on.nil?
     journal.details << JournalDetail.new(:property => 'relation',
                                          :prop_key => relation_type_for(issue_to),
                                          :value    => issue_from.id)
```

I thought the IssueRelation class could be improved so that it works less on assumptions, in order to make it usable from a migration task.

Associated revisions

Revision 13534 - 2014-11-02 16:38 - Jean-Philippe Lang

Fixed that IssueRelation should not be responsible for calling Issue#init_journal (#18237).

History

#1 - 2014-10-30 09:06 - Jean-Philippe Lang

- Assignee set to Jean-Philippe Lang

- Target version set to 3.0.0

#2 - 2014-11-02 16:40 - Jean-Philippe Lang

- Category changed from Issues to Code cleanup/refactoring

- Status changed from New to Closed

- Resolution set to Fixed

Fixed in r13534. Adding a relation won't create a journal, unless Issue#init_journal or IssueRelation#init_journals was called.

Example:

```
relation = IssueRelation.new attributes
relation.save # no journals created

relation = IssueRelation.new attributes
relation.init_journals(some_user)
relation.save # journals created
```

#3 - 2014-11-02 22:41 - Stephane Lapie

Thanks for the fix !

Actually, now, the relation.init_journals method ends up implicitely returning only the new journal made in "issue_to".
This is neat as it would now enable me to modify the creation time, which for my current use case is quite important : I am migrating data from another ticketing system, and I want to preserve history and original dates. However, this patch only allows to do this for "issue_to", out of the two journal entries.

Could it be possible to patch app/models/issue_relation.rb in the following way?

```
  def init_journals(user)
    journals = {}
    journals[:issue_from] = issue_from.init_journal(user) if issue_from
    journals[:issue_to] = issue_to.init_journal(user) if issue_to
    journals
  end
```

That way, one could access both journal items after they have been commited once, and modify them again if needed.

I am aware this is a bit dirty and probably an unintended use, but then is the better option to manually create the journal entries on my own?

Thanks in advance,