For now, in 10-patches.rb we have a redefinition of method `human_attribute_name`. But default (plain without hierarchy) Redmine style of transliterations are not convinient for some reasons, especially in big projects. So, my proposal is follows: enable ActiveRecord behaviour for finding transliterations path and add in list of search paths custom one from Redmine. Here is a worked snippet of code:

```ruby
def self.human_attribute_name(attr, options = {})
  prepared_attr = attr.to_s.sub(/_id$/, '').sub(/[^.]+/, '')

  redmine_default =
  [:field_#{name.underscore.gsub('/', '_').downcase},
   :field_#{prepared_attr}]

  if options[:default].present?
    options[:default] = [options[:default]] unless options[:default].is_a? Array
    options[:default].unshift redmine_default
  else
    options[:default] = redmine_default
  end

  super
end
```

In this case we obtain follows chain of search paths:
1. default AR paths (for example: activerecord.attributes.model_name.field_name)
2. default Redmine paths (for example: field_content)
3. user specified paths (specified in `:default` key from `options` attribute)
4. attribute.humanize (from default `human_attribute_name` from ActiveModel)
Code cleanup (#22383).

History

#1 - 2016-04-09 23:54 - Jean-Philippe Lang
- Target version set to 3.3.0

#2 - 2016-04-12 21:15 - Jean-Philippe Lang
- Status changed from New to Closed
- Assignee set to Jean-Philippe Lang
- Resolution set to Fixed

Change committed, thanks.

#3 - 2016-04-30 17:53 - Toshi MARUYAMA
- Description updated

#4 - 2018-06-04 04:24 - Go MAEDA
- Related to Feature #26618: Support of default ActiveRecord I18n scopes in LabelledFormBuilder added