Ruby trunk - Feature #7113

New classes: `HumanTime::LocalTime`, `HumanTime::LocalDate`, `HumanTime::LocalDateTime` to represent "timetable time"

10/07/2012 12:05 AM - alexeymuranov (Alexey Muranov)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		

Description

=begin
I think it would be nice if the following new classes were added to the standard library: (({HumanTime::LocalTime})), (({HumanTime::LocalDate})), (({HumanTime::LocalDateTime})) to represent what they name. Apart from the time-related instance variables (({@year})), (({@month})), (({@day})), (({@weekday})), (({@hour})), (({@minutes})), (({@seconds})), (({@fraction_of_a_second})), they would have an optional string instance variable (({@location})), and maybe (({@longitude})) and (({@latitude})).

I think that the (({Time})) class is good for timestamping system log entires or stages in a physical experiment, but is not appropriate for "timetable time", like train schedule.

Here is an example: if the Mood falls on Earth and the Earth starts rotating faster, the days would become shorter, and it is clear that trains would not continue to follow their previous schedule described with Ruby (({Time})) instances, but they will try to follow the schedule described with "hh:mm" format relative to the new day length, which will stay 24 ((*human hours*)).

Another practical consideration: Ruby currently does not have a class to represent a standard database "time" column. This causes some unpleasant difficulties to users of ((*Ruby on Rails*)) web application framework, for example:
http://www.rebeccablyth.co.uk/2008/01/03/time-columns-and-rails/

What do you think?
=end

History

#1 - 10/07/2012 12:15 AM - alexeymuranov (Alexey Muranov)

I forgot one more: HumanTime::Duration.

#2 - 10/07/2012 04:43 PM - alexeymuranov (Alexey Muranov)

Sorry, there was a typo: *Moon*, of course :).

#3 - 10/07/2012 05:28 PM - drbrain (Eric Hodel)

- Status changed from Open to Rejected

I don't see the need for thr inclusion of a HumanTime class as you describe in ruby. Typically a time with time zone is sufficient (time zones were popularized by railroads to add convenience to scheduling) which ruby provides (although, conversion between zones can be cumbersome).

You should make a gem to hold the classes you describe instead. If many rubyists need and use your gem you can nominate it for inclusion in ruby.

Please file a separate issue for your time-of-day class. Issues need to cover only one bug or feature suggestion.

PS: The Moon cannot impact the earth. Time-keeping systems deal with the drift of the length of the day in a manner that doesn't impact train or other schedules. Even if the Moon could impact the earth there wouldn't be any trains (or people, or animals) left to schedule anything (a 15km asteroid will create a mass-extinction level impact, the Moon is much larger and would liquefy the Earth's crust, including all the railroads).

#4 - 10/08/2012 02:11 AM - alexeymuranov (Alexey Muranov)

Eric, i meant that measuring the time in seconds ("the duration of 9 192 631 770 periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the cesium 133 atom", according to the internet) from Epoch (January 1, 1970 00:00 UTC) does not look to me like a natural way to store train timetable. I think that avoiding division to convert this number to the day time each time can give some performance increase (or maybe save energy :)).

P.S. I do not think that the Three-Body Problem (here: Sun - Earth - Moon) is sufficiently studied to exclude the possibility that Moon *eventually* falls on Earth (here is what i've found with google: http://count.ucsc.edu/~rmont/papers/S10.pdf). Also Moon can be pulled closer gently with a rope to avoid an impact.

P.P.S. I will think about a gem.

Update 2012-10-12: I think that the relation between Time and HumanTime::LocalDateTime would be somewhat like the relation between Rational and Float (Float can also be thought of as redundant).