carbon_model.rb

Automobile carbon model

This model is used by Brighter Planet's carbon emission web service to estimate the **greenhouse gas emissions of an automobile**.

Timeframe, acquisition, and retirement

The model estimates the emissions that occur during a particular `timeframe`. To do this it needs to know the automobile's `acquisition` (the date it started being used) and `retirement` (the date it stopped being used). For example, if the `timeframe` is January 2010, an automobile with `acquisition` of January 2009 and `retirement` of February 2010 will have emissions, but an automobile with `acquisition` of February 2010 or `retirement` of December 2009 will not.

Calculations

The final estimate is the result of the **calculations** detailed below. These calculations are performed in reverse order, starting with the last calculation listed and finishing with the `emission` calculation. Each calculation is named according to the value it returns.

Methods

To accomodate varying client input, each calculation may have one or more **methods**. These are listed under each calculation in order from most to least preferred. Each method is named according to the values it requires. If any of these values is not available the method will be ignored. If all the methods for a calculation are ignored, the calculation will not return a value. "Default" methods do not require any values, and so a calculation with a default method will always return a value.

Standard compliance

```
require 'conversions'


module BrighterPlanet
  module Automobile
    module CarbonModel
      def self.included(base)
        base.decide :emission, :with => :characteristics do
```

Each method lists any established calculation standards with which it **complies**. When compliance with a standard is requested, all methods that do not comply with that standard are ignored. This means that any values a particular method requires will have been calculated using a compliant method, because those are the only methods available. If any value did not have a compliant method in its calculation then it would be undefined, and the current method would have been ignored.

Collaboration

Contributions to this carbon model are actively encouraged and warmly welcomed. This library includes a comprehensive test suite to ensure that your changes do not cause regressions. All changes should include test coverage for new functionality. Please see sniff, our emitter testing framework, for more information.

Emission calculation

Returns the `emission` estimate ($kg\ CO_2e$).

Emission from CO_2 emission, CH_4 emission, N_2O emission, and HFC emission

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Sums the non-biogenic emissions to give $kg\ CO_2e$.

Emission from default

Displays an error message if the previous method fails.

CO_2 emission calculation

```
          committee :emission do


            quorum 'from co2 emission, ch4 emission, n2o emission, and hfc emission',
              :needs => [:co2_emission, :ch4_emission, :n2o_emission, :hfc_emission],

              :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|

                characteristics[:co2_emission] + characteristics[:ch4_emission] +
characteristics[:n2o_emission] + characteristics[:hfc_emission]
            end


            quorum 'default' do

              raise "The emission committee's default quorum should never be called"
            end
          end


          committee :co2_emission do
```

Returns the `co2 emission` ($kg\ CO_2$).

CO_2 emission from fuel use and CO_2 emission factor

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Multiplies `fuel use` (l) by the `co2 emission factor` ($kg\ CO_2 / l$) to give $kg\ CO_2$.

```
            quorum 'from fuel use and co2 emission factor',
              :needs => [:fuel_use, :co2_emission_factor],

              :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|

                characteristics[:fuel_use] * characteristics[:co2_emission_factor]
            end
          end
```

CO_2 biogenic emission calculation

Returns the `co2 biogenic emission` ($kg\ CO_2$).

CO_2 biogenic emission from fuel use and CO_2 biogenic emission factor

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Multiplies `fuel use` (l) by the `co2 biogenic emission factor` ($kg\ CO_2 / l$) to give $kg\ CO_2$.

```
          committee :co2_biogenic_emission do

            quorum 'from fuel use and co2 biogenic emission factor',
              :needs => [:fuel_use, :co2_biogenic_emission_factor],

              :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|

                characteristics[:fuel_use] *
characteristics[:co2_biogenic_emission_factor]
            end
          end
```

CH_4 emission calculation

Returns the `ch4 emission` ($kg\ CO_2e$).

CH_4 emission from fuel use and CH_4 emission factor

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

```
          committee :ch4_emission do

            quorum 'from fuel use and ch4 emission factor',
              :needs => [:fuel_use, :ch4_emission_factor],

              :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|
```

Multiplies `fuel use` (l) by the `ch4 emission factor` ($kg\ CO_2e\ /\ l$) to give $kg\ CO_2e$.

N_2O emission calculation

Returns the `n2o emission` ($kg\ CO_2e$).

N_2O emission from fuel use and N_2O emission factor

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Multiplies `fuel use` (l) by the `n2o emission factor` ($kg\ CO_2e\ /\ l$) to give $kg\ CO_2e$.

HFC emission calculation

Returns the `hfc emission` ($kg\ CO_2e$).

HFC emission from fuel use and HFC emission factor

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Multiplies `fuel use` (l) by the `hfc emission factor` ($kg\ CO_2e\ /\ l$) to give $kg\ CO_2e$.

CO_2 emission factor calculation

Returns the `co2 emission factor` ($kg\ /\ l$).

```
            characteristics[:fuel_use] * characteristics[:ch4_emission_factor]
          end
        end


        committee :n2o_emission do


          quorum 'from fuel use and n2o emission factor',
            :needs => [:fuel_use, :n2o_emission_factor],

            :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|

            characteristics[:fuel_use] * characteristics[:n2o_emission_factor]
          end
        end


        committee :hfc_emission do


          quorum 'from fuel use and hfc emission factor',
            :needs => [:fuel_use, :hfc_emission_factor],

            :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|

            characteristics[:fuel_use] * characteristics[:hfc_emission_factor]
          end
        end


        committee :co2_emission_factor do
```

CO_2 emission factor from automobile fuel

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Looks up the fuel `co2 emission factor` (kg / l).

CO_2 biogenic emission factor calculation

Returns the `co2 biogenic emission factor` (kg / l).

CO_2 biogenic emission factor from automobile fuel

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Looks up the fuel `co2 biogenic emission factor` (kg / l).

CH_4 emission factor calculation

Returns the `ch4 emission factor` ($kg\ CO_2e / l$).

CH_4 emission factor from automobile fuel

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Looks up the fuel `ch4 emission factor` ($kg\ CO2e / l$).

```
            quorum 'from automobile fuel',
              :needs => :automobile_fuel,

              :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|

                characteristics[:automobile_fuel].co2_emission_factor
              end
            end


          committee :co2_biogenic_emission_factor do



            quorum 'from automobile fuel',
              :needs => :automobile_fuel,

              :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|

                characteristics[:automobile_fuel].co2_biogenic_emission_factor
              end
            end


          committee :ch4_emission_factor do



            quorum 'from automobile fuel',
              :needs => :automobile_fuel,

              :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|

                characteristics[:automobile_fuel].ch4_emission_factor
              end
```

N_2O emission factor calculation

Returns the `n2o emission factor` ($kg\ CO_2e\ /\ l$).

N_2O emission factor from automobile fuel

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Looks up the fuel `n2o emission factor` ($kg\ CO2e\ /\ l$).

HFC emission factor calculation

Returns the `hfc emission factor` ($kg\ CO_2e\ /\ l$).

HFC emission factor from automobile fuel

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Looks up the fuel `hfc emission factor` ($kg\ CO2e\ /\ l$).

Fuel use calculation

Returns the trip `fuel use` (l).

Fuel use from fuel efficiency and distance

```
          end


          committee :n2o_emission_factor do



            quorum 'from automobile fuel',
              :needs => :automobile_fuel,

              :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|

                characteristics[:automobile_fuel].n2o_emission_factor
            end
          end


          committee :hfc_emission_factor do



            quorum 'from automobile fuel',
              :needs => :automobile_fuel,

              :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|

                characteristics[:automobile_fuel].hfc_emission_factor
            end
          end


          committee :fuel_use do



            quorum 'from fuel efficiency and distance',
              :needs => [:fuel_efficiency, :distance],
```

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Divides the `distance` (km) by the `fuel efficiency` (km / l) to give l.

```
            :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|

              characteristics[:distance] / characteristics[:fuel_efficiency]
            end
          end
```

Distance calculation

Returns the `distance` (km). This is the distance the automobile travelled during the `active subtimeframe`.

```
          committee :distance do
```

Distance from annual distance

Complies: GHG Protocol Scope 3, ISO 14064-1

```
            quorum 'from annual distance',
              :needs => [:annual_distance, :active_subtimeframe],
```

Multiplies the `annual distance` (km) by the fraction of the calendar year in which the `timeframe` falls that overlaps with the `active subtimeframe`.

```
              :complies => [:ghg_protocol_scope_3, :iso] do |characteristics,
timeframe|

              characteristics[:annual_distance] *
(characteristics[:active_subtimeframe] / timeframe.year)
            end
          end
```

Annual distance calculation

Returns the `annual distance` (km). This is the distance the automobile would travel if it were in use for the entire calendar year in which the `timeframe` falls. Note that if either `acquisition` or `retirement` occurs during the calendar year in which the `timeframe` falls then `annual distance` will be MORE THAN the distance the automobile actually travelled during that calendar year.

```
          committee :annual_distance do
```

Annual distance from client input

Complies: All

Uses the client-input `annual distance` (km).

Annual distance from weekly distance and timeframe

Complies: GHG Protocol Scope 3, ISO 14064-1

Divides the `weekly distance` (*km*) by 7 and multiplies by the number of days in the calendar year in which the `timeframe` falls.

Annual distance from daily distance and timeframe

Complies: GHG Protocol Scope 3, ISO 14064-1

Multiplies the `daily distance` (*km*) by the number of days in the calendar year in which the `timeframe` falls.

Annual distance from daily duration, speed, and timeframe

Complies: GHG Protocol Scope 3, ISO 14064-1

Multiplies the `daily duration` (*seconds*) by the `speed` (*km / hour*) to give *km*. Multiplies the result by the number of days in the calendar year in which the `timeframe` falls.

Annual distance from size class

Complies: GHG Protocol Scope 3, ISO 14064-1

Looks up the automobile size class `annual distance` (*km*).

```
            quorum 'from weekly distance and timeframe',
              :needs => :weekly_distance,

              :complies => [:ghg_protocol_scope_3, :iso] do |characteristics,
timeframe|

                (characteristics[:weekly_distance] / 7 ) * timeframe.year.days
            end


            quorum 'from daily distance and timeframe',
              :needs => :daily_distance,

              :complies => [:ghg_protocol_scope_3, :iso] do |characteristics,
timeframe|

                characteristics[:daily_distance] * timeframe.year.days
            end


            quorum 'from daily duration, speed, and timeframe',
              :needs => [:daily_duration, :speed],

              :complies => [:ghg_protocol_scope_3, :iso] do |characteristics,
timeframe|

                characteristics[:daily_duration] / 3600.0 * characteristics[:speed] *
timeframe.year.days
            end


            quorum 'from size class',
              :needs => :size_class,

              :complies => [:ghg_protocol_scope_3, :iso] do |characteristics|

                characteristics[:size_class].annual_distance
            end
```

Annual distance from automobile fuel

Complies: GHG Protocol Scope 3, ISO 14064-1

Looks up the automobile fuel `annual distance` (*km*).

```
            quorum 'from automobile fuel',
              :needs => :automobile_fuel,

              :complies => [:ghg_protocol_scope_3, :iso] do |characteristics|

                characteristics[:automobile_fuel].annual_distance
              end
            end
```

Weekly distance calculation

Returns the client-input `weekly distance` (*km*). This is the average distance the automobile travels each week.

Daily distance calculation

Returns the client-input `daily distance` (*km*). This is the average distance the automobile travels each day.

Daily duration calculation

Returns the client-input `daily duration` (*seconds*).

Automobile fuel calculation

Returns the type of `automobile fuel` used.

```
          committee :automobile_fuel do
```

Automobile fuel from client input

Complies: All

Uses the client-input automobile fuel.

Automobile fuel from make model year variant

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

```
            quorum 'from make model year variant',
              :needs => :make_model_year_variant,

              :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|
```

Looks up the variant `automobile fuel`.

Default automobile fuel

Complies: GHG Protocol Scope 3, ISO 14064-1

Looks up the default automobile fuel.

Speed calculation

Returns the average `speed` at which the automobile travels ($km / hour$).

Speed from client input

Complies: All

Uses the client-input `speed` ($km / hour$).

Speed from urbanity

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Takes average city and highway driving speeds from EPA (2006) and converts from $miles / hour$ to $km / hour$, then calculates the harmonic mean of those speeds weighted by `urbanity`.

Fuel efficiency calculation

Returns the `fuel efficiency` (km / l)

```
            characteristics[:make_model_year_variant].fuel
          end


          quorum 'default',

            :complies => [:ghg_protocol_scope_3, :iso] do |characteristics|

            AutomobileFuel.fallback
          end
        end


        committee :speed do


          quorum 'from urbanity',
            :needs => :urbanity,

            :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|

              1 / (characteristics[:urbanity] / base.fallback.city_speed + (1 -
characteristics[:urbanity]) / base.fallback.highway_speed)
          end
        end


        committee :fuel_efficiency do
```

Fuel efficiency from client input

Complies: All

Uses the client-input `fuel efficiency` (km/l).

Fuel efficiency from make model year variant and urbanity

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Looks up the city and highway fuel efficiencies of the automobile make model year variant (km/l).

Calculates the harmonic mean of those fuel efficiencies, weighted by `urbanity`.

Fuel efficiency from make model year and urbanity

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Looks up the city and highway fuel efficiencies of the automobile make model year (km/l).

Calculates the harmonic mean of those fuel efficiencies, weighted by `urbanity`.

```
            quorum 'from make model year variant and urbanity',
              :needs => [:make_model_year_variant, :urbanity],

              :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|

                fuel_efficiency_city =
characteristics[:make_model_year_variant].fuel_efficiency_city
                fuel_efficiency_highway =
characteristics[:make_model_year_variant].fuel_efficiency_highway
                urbanity = characteristics[:urbanity]
                if fuel_efficiency_city.present? and fuel_efficiency_highway.present?

                  1.0 / ((urbanity / fuel_efficiency_city) + ((1.0 - urbanity) /
fuel_efficiency_highway))
                end
            end


            quorum 'from make model year and urbanity',
              :needs => [:make_model_year, :urbanity],

              :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|

                fuel_efficiency_city =
characteristics[:make_model_year].fuel_efficiency_city
                fuel_efficiency_highway =
characteristics[:make_model_year].fuel_efficiency_highway
                urbanity = characteristics[:urbanity]
                if fuel_efficiency_city.present? and fuel_efficiency_highway.present?

                  1.0 / ((urbanity / fuel_efficiency_city) + ((1.0 - urbanity) /
fuel_efficiency_highway))
                end
            end
```

Fuel efficiency from make model and urbanity

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Looks up the city and highway fuel efficiencies of the automobile make model (km / l).

Calculates the harmonic mean of those fuel efficiencies, weighted by `urbanity`.

```
            quorum 'from make model and urbanity',
              :needs => [:make_model, :urbanity],

              :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|

                fuel_efficiency_city =
characteristics[:make_model].fuel_efficiency_city
                fuel_efficiency_highway =
characteristics[:make_model].fuel_efficiency_highway
                urbanity = characteristics[:urbanity]
                if fuel_efficiency_city.present? and fuel_efficiency_highway.present?

                  1.0 / ((urbanity / fuel_efficiency_city) + ((1.0 - urbanity) /
fuel_efficiency_highway))
                end
            end
```

Fuel efficiency from size class, hybridity multiplier, and urbanity

Complies: GHG Protocol Scope 3, ISO 14064-1

Looks up the automobile size class city and highway fuel efficiency (km / l).

Calculates the harmonic mean of those fuel efficiencies, weighted by `urbanity`, and multiplies the result by the `hybridity multiplier`.

```
            quorum 'from size class, hybridity multiplier, and urbanity',
              :needs => [:size_class, :hybridity_multiplier, :urbanity],

              :complies => [:ghg_protocol_scope_3, :iso] do |characteristics|

                fuel_efficiency_city =
characteristics[:size_class].fuel_efficiency_city
                fuel_efficiency_highway =
characteristics[:size_class].fuel_efficiency_highway
                urbanity = characteristics[:urbanity]
                if fuel_efficiency_city.present? and fuel_efficiency_highway.present?

                  (1.0 / ((urbanity / fuel_efficiency_city) + ((1.0 - urbanity) /
fuel_efficiency_highway))) * characteristics[:hybridity_multiplier]
                end
            end
```

Fuel efficiency from make year and hybridity multiplier

```
            quorum 'from make year and hybridity multiplier',
              :needs => [:make_year, :hybridity_multiplier],
```

Complies: GHG Protocol Scope 3, ISO 14064-1

Looks up the automobile make year combined fuel efficiency (km/l) and multiplies it by the `hybridity multiplier`.

Fuel efficiency from make and hybridity multiplier

Complies: GHG Protocol Scope 3, ISO 14064-1

Looks up the automobile make combined fuel efficiency (km/l) and multiplies it by the `hybridity multiplier`.

Fuel efficiency from hybridity multiplier

Complies: GHG Protocol Scope 3, ISO 14064-1

Takes a default `fuel efficiency` of 8.58 km/l, calculated from total US automobile vehicle miles travelled and gasoline and diesel use, and multiplies it by the `hybridity multiplier`.

Size class calculation

Returns the client-input automobile size class.

Hybridity multiplier calculation

Returns the `hybridity multiplier`. This value may be used to adjust the fuel efficiency based on whether the automobile is a hybrid or conventional vehicle.

```
            :complies => [:ghg_protocol_scope_3, :iso] do |characteristics|

              characteristics[:make_year].fuel_efficiency *
characteristics[:hybridity_multiplier]
            end



            quorum 'from make and hybridity multiplier',
              :needs => [:make, :hybridity_multiplier],

              :complies => [:ghg_protocol_scope_3, :iso] do |characteristics|

              if characteristics[:make].fuel_efficiency.present?
                characteristics[:make].fuel_efficiency *
characteristics[:hybridity_multiplier]
              end
            end



            quorum 'from hybridity multiplier',
              :needs => :hybridity_multiplier,

              :complies => [:ghg_protocol_scope_3, :iso] do |characteristics|

              base.fallback.fuel_efficiency * characteristics[:hybridity_multiplier]
            end
          end




          committee :hybridity_multiplier do
```

Hybridity multiplier from size class, hybridity, and urbanity

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Looks up the appropriate city and highway hybridity multipliers for the automobile size class.

Calculates the harmonic mean of those multipliers, weighted by `urbanity`.

```
            quorum 'from size class, hybridity, and urbanity',
              :needs => [:size_class, :hybridity, :urbanity],


              :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|

                drivetrain = characteristics[:hybridity] ? :hybrid : :conventional
                urbanity = characteristics[:urbanity]
                size_class = characteristics[:size_class]
                fuel_efficiency_multipliers = {
                  :city =>
size_class.send(:"#{drivetrain}_fuel_efficiency_city_multiplier"),
                  :highway =>
size_class.send(:"#{drivetrain}_fuel_efficiency_highway_multiplier")
                }
                if fuel_efficiency_multipliers.values.any?(&:present?)

                  1.0 / ((urbanity / fuel_efficiency_multipliers[:city]) + ((1.0 -
urbanity) / fuel_efficiency_multipliers[:highway]))
                else
                  nil
                end
            end
```

Hybridity multiplier from hybridity and urbanity

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Looks up the appropriate default city and highway hybridity multipliers.

Calculates the harmonic mean of those multipliers, weighted by `urbanity`.

```
            quorum 'from hybridity and urbanity',
              :needs => [:hybridity, :urbanity],


              :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics|

                drivetrain = characteristics[:hybridity] ? :hybrid : :conventional
                urbanity = characteristics[:urbanity]
                fuel_efficiency_multipliers = {
                  :city =>
AutomobileSizeClass.fallback.send(:"#{drivetrain}_fuel_efficiency_city_multiplier"),
                  :highway =>
AutomobileSizeClass.fallback.send(:"#{drivetrain}_fuel_efficiency_highway_multiplier")
                }

                1.0 / ((urbanity / fuel_efficiency_multipliers[:city]) + ((1.0 -
urbanity) / fuel_efficiency_multipliers[:highway]))
```

Default hybridity multiplier

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Uses a default `hybridity multiplier` of 1.

```
  end


  quorum 'default',

    :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do

      base.fallback.hybridity_multiplier
  end
end
```

Hybridity calculation

Returns the client-input `hybridity`. This indicates whether the automobile is a hybrid electric vehicle or a conventional vehicle.

Urbanity calculation

Returns the `urbanity`. This is the fraction of the total distance driven that occurs on towns and city streets as opposed to highways (defined using a 45 miles per hour "speed cutpoint").

```
committee :urbanity do
```

Default urbanity

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Uses an `urbanity` of 0.43 after EPA (2009) Appendix A .

```
  quorum 'default',

    :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do

      base.fallback.urbanity
  end
end
```

Active subtimeframe calculation

Returns the portion of the `timeframe` that falls between the `acquisition` and `retirement`.

```
committee :active_subtimeframe do
```

Active subtimeframe from timeframe, acquisition, and retirement

```
  quorum 'from acquisition and retirement',
    :needs => [:acquisition, :retirement],
```

Complies: GHG Protocol Scope 1, GHG Protocol Scope 3, ISO 14064-1

Uses the portion of the `timeframe` that falls between `acquisition` and `retirement`.

Acquisition calculation

Returns the date of the automobile's `acquisition`. This is the date the automobile was put into use.

Acquisition from client input

Complies: All

Uses the client-input `acquisition`.

Acquisition from make model year variant

Complies: GHG Protocol Scope 3, ISO 14064-1

Uses the first day of the client-input automobile make model year variant year.

Acquisition from make model year

Complies: GHG Protocol Scope 3, ISO 14064-1

Uses the first day of the client-input automobile make model year year.

Acquisition from make year

```
              :complies => [:ghg_protocol_scope_1, :ghg_protocol_scope_3, :iso] do
|characteristics, timeframe|

                Timeframe.constrained_new characteristics[:acquisition].to_date,
characteristics[:retirement].to_date, timeframe
              end
            end


            committee :acquisition do









              quorum 'from make model year variant',
                :needs => [:make_model_year_variant],

                :complies => [:ghg_protocol_scope_3, :iso] do |characteristics|

                  Date.new characteristics[:make_model_year_variant].year, 1, 1
                end


              quorum 'from make model year',
                :needs => [:make_model_year],

                :complies => [:ghg_protocol_scope_3, :iso] do |characteristics|

                  Date.new characteristics[:make_model_year].year, 1, 1
                end


              quorum 'from make year',
```

Complies: GHG Protocol Scope 3, ISO 14064-1

Uses the first day of the client-input automobile make year year.

Acquisition from timeframe or retirement

Complies: GHG Protocol Scope 3, ISO 14064-1

Uses the first day of the `timeframe`, or the `retirement`, whichever is earlier.

Retirement calculation

Returns the date of the automobile's `retirement`. This is the date the automobile was taken out of use.

Retirement from client input

Complies: All

Uses the client-input `retirement`.

Retirement from timeframe or acquisition

Complies: GHG Protocol Scope 3, ISO 14064-1

Uses the last day of the `timeframe`, or the `acquisition`, whichever is later.

```
              :needs => [:make_year],

              :complies => [:ghg_protocol_scope_3, :iso] do |characteristics|

                Date.new characteristics[:make_year].year, 1, 1
            end


            quorum 'from retirement',
              :appreciates => :retirement,

              :complies => [:ghg_protocol_scope_3, :iso] do |characteristics,
timeframe|

                [ timeframe.from, characteristics[:retirement] ].compact.min
            end
          end


          committee :retirement do









            quorum 'from acquisition',
              :appreciates => :acquisition,

              :complies => [:ghg_protocol_scope_3, :iso] do |characteristics,
timeframe|

                [ timeframe.to, characteristics[:acquisition] ].compact.max
            end
          end
```

Make model year variant calculation

Returns the client-input automobile make model year variant.

Make model year calculation

Returns the client-input automobile make model year.

Make model calculation

Returns the client-input automobile make model.

Make year calculation

Returns the client-input automobile make year.

Make calculation

Returns the client-input automobile make.

Timeframe calculation

Returns the `timeframe`. This is the period during which to calculate emissions.

Timeframe from client input

Complies: All

Uses the client-input `timeframe`.

Default timeframe

Complies: All

Uses the current calendar year.

```
        end
      end
    end
  end
end
```