Learn to Create Better Models with Units of Measurement

September 17, 2014

Chris Kuip

Testimonials from our partners

For the **end-user**:

"*Problems involving lots of entities tend to require **different** units, even within the same dimension. For instance, in OCC some SKU are input as "ton" while others are input as kg or even g. It is more natural than requiring users to do **manual** conversion (like **forcing** them to convert everything to kg in Excel before input). Internally we can have identifiers in ton for example but show them in kg in reports (by adjusting pivot table settings). AIMMS does all the **conversion** for us.*"

For the **developer**:

"*We are **free** to model with whatever unit we want. If the user asks us to change units in GUI, it's just a visualization matter. As the model grows, having all identifiers with units helps avoid **inconsistencies** in expressions and constraints. For example, if a parameter that expects a "ton/day" value receives "ton" AIMMS raises a warning, so we can fix the assignment and/or its units. As AIMMS does all the conversion for us, we don't need to worry. I can **happily** assign a "kg" value to a "ton" parameter and it just works.*"

Units of Measurement (UoM), intro

- UoM give meaning to data, compare:
  - 15
  - 15 [m]
- On the globe several different UoM systems are known. The two most common ones are:
  - Imperial, used in trade, UK, and USA
  - Metric, used in scientific world, Europe, Asia, and Australia

- Obviously, having multiple UoM systems leads to, costly, confusions:

- In 1999 the Mars Climate Orbiter, worth 125M$, crashed.
- Final analysis report was signed by 10 managers at NASA.
- Root cause:
  Inconsistent use of UoM communicating some small but significant forces between selected software components.
- Recommendation:
  Verify the consistent use of units throughout the spacecraft design and operations.

Intro Question

Please take on the role of program manager. You want to avoid disasters such as the one just described. For a new project, what do you choose?

1. Everything Imperial, most USA citizens are used to it
2. Everything Metric, scientists, most Europeans are used to it
3. Up to each developer, ensure proper documentation
4. For each user, allow customization
5. Decide per project one of the above, as appropriate

Poll: think about it for a minute

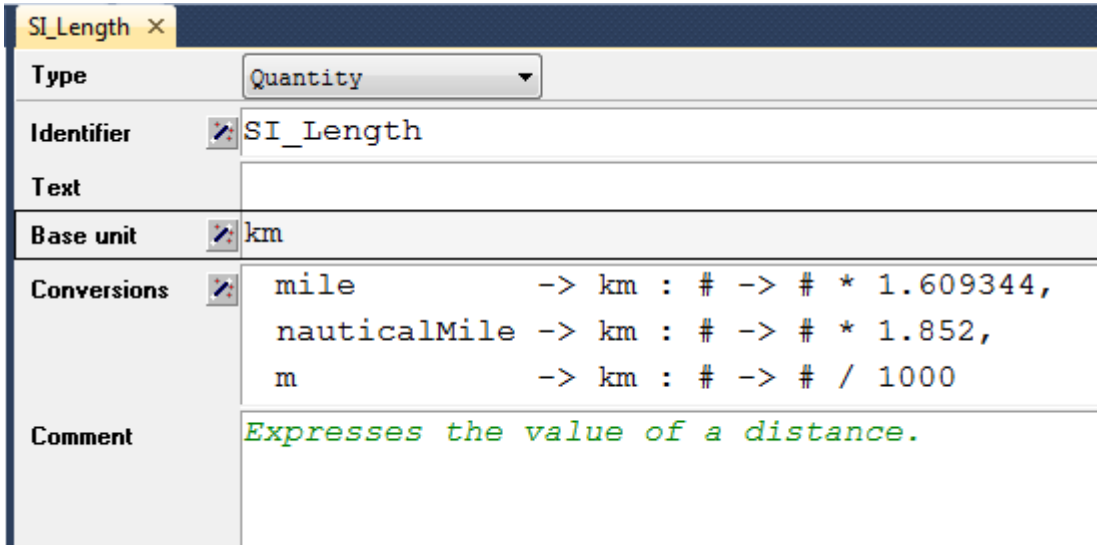
By now you probably realize this is a trick question.

Claim: with AIMMS, no need to make this choice:
Every stake holder can use customary UoM

Demo 1: Declaring units.

- Quantity (what do we measure, which units do we use):
  - Create quantity SI_Length
  - Add conversions for miles and km by point and click
  - Add conversion for nautical miles, factor 1852, by editing attribute
  - Switch to basic unit km (model granularity)

    Designing a circuit board has a different base unit than flight planning.



Demo 1 continued: Associating units

- Add unit mile to an existing parameter

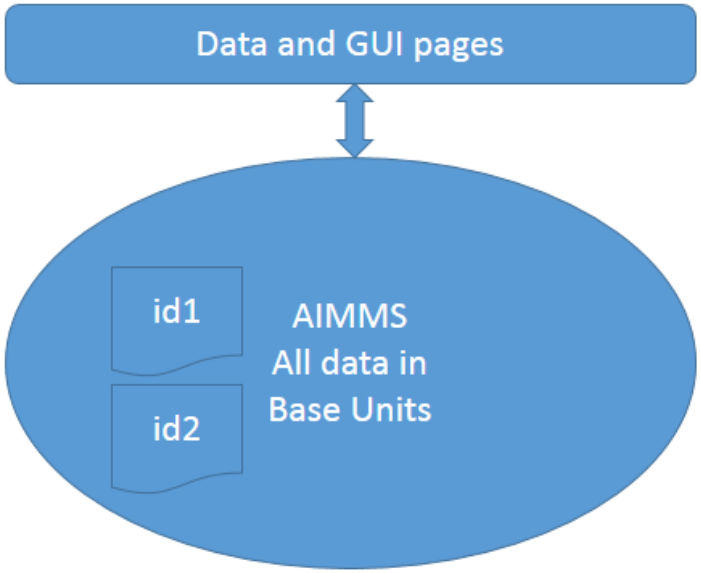
```
Parameter DistanceTable {
    IndexDomain: (i,j);
    Unit: mile;
    InitialData: {
        data table
                    London     Paris   NewYork
                 !  -------   -------  -------
           Paris       213
         NewYork      3465      3631
        Shanghai      5721      5774     7377
```

- Show data page, with units per value in nauticalMile

[Data Page] DistanceTable

j / i	London	Paris	NewYork	Shanghai
London				
Paris	185.0919395[nauticalMile]			
NewYork	3011.002678[nauticalMile]	3155.252734[nauticalMile]		
Shanghai	4971.413080[nauticalMile]	5017.468821[nauticalMile]	6410.437737[nauticalMile]	

Units of measurement – AIMMS under the hood



The bi-directional arrow handles the conversion of numeric data according to the units of measurement selected.

- AIMMS stores all numeric data according to the base units selected for the quantities
- Whenever AIMMS communicates numeric data it will convert that data according to the selected units

Conventions

- A group of people with a common background, use the same UoM for measuring. For instance:
  - Europeans: distance between cities, by convention, in km
  - USA citizens: distance between cities, by convention, in miles
- A convention is an agreed upon collection of units, according to which numeric data is viewed, typical examples:
  - `convImperial`, units such as pound, gallon, second, and yard
  - `convMetric`, units such as kg, liter, second, and m

Demo 2: Adding conventions

- The convention `conv Imperial`

```
Convention conv_Imperial {
    PerQuantity: {
        SI_Length        : yd,
        SI_Mass          : lb,
        SI_Time_Duration : s,
        SI_Volume        : gal
```

- The element parameter `ep_currConv`

```
ElementParameter ep_currConv {
    Range: AllConventions;
    InitialData: 'conv_Metric';
```

- The attributes of the main model

```
Model Demo_2 {
    Convention: ep_currConv;
```

Demo 2 continued: Using conventions

- The distance table according to both conventions:

DistanceTable
[km]

	London	Paris	NewYork	Shanghai
London				
Paris	343			
NewYork	5576	5844		
Shanghai	9207	9292	11872	

ep_currConv = conv_Metric

DistanceTable
[mile]

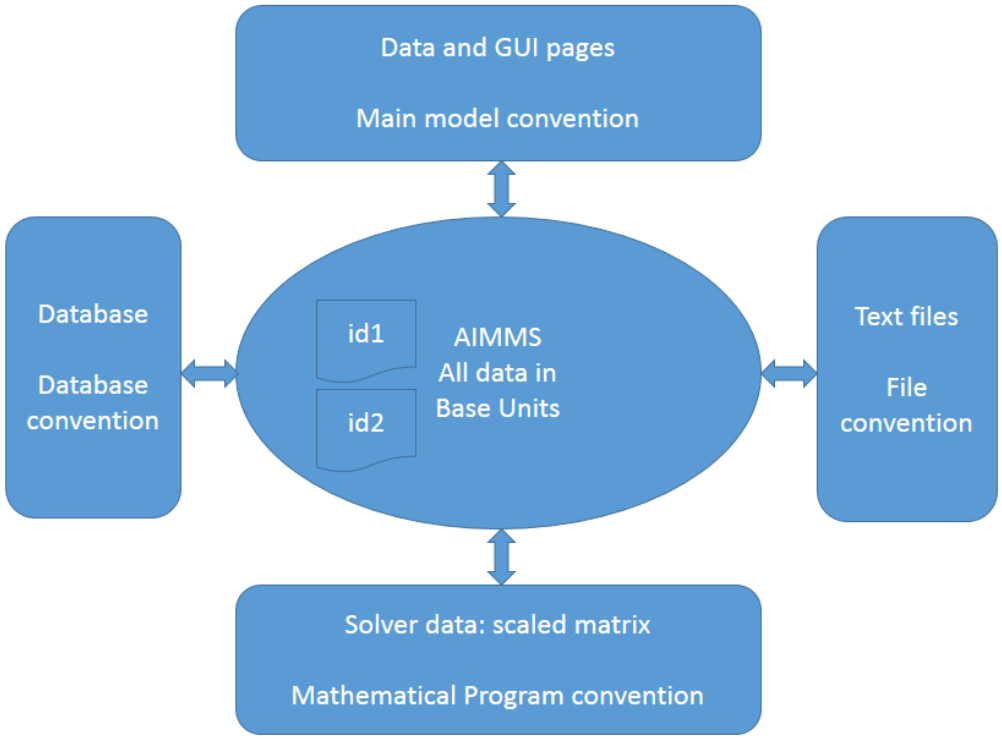
	London	Paris	NewYork	Shanghai
London				
Paris	213			
NewYork	3465	3631		
Shanghai	5721	5774	7377	

ep_currConv = conv_Imperial

- Paris – London
  343 [km] or 213 [mile]
- Note the use of the unit in the header
- Control via ep_currConv

Conversions according to conventions

- All communication channels:



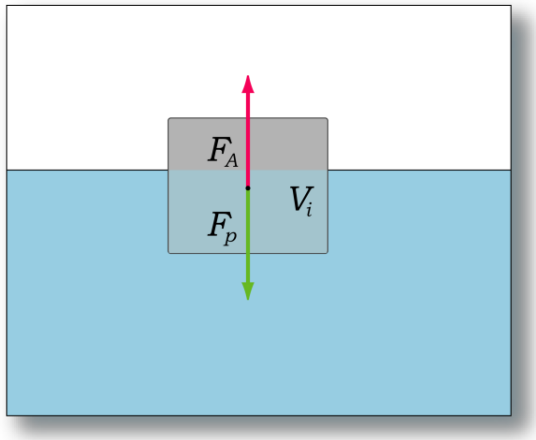
The bi-directional arrow handles the conversion of numeric data according to the units of measurement selected by a convention.

AIMMS' interaction on numeric data can also go via text input files and via databases. There is also interaction with solvers. In all of these communication channels, the selection of units can be controlled via conventions

Unit consistency, poll

Archimedes law:

- weightBody in [kg]
- volumeBody in [m3]
- densityFluid in [kg/m3]



Possible formulas for afloat test:

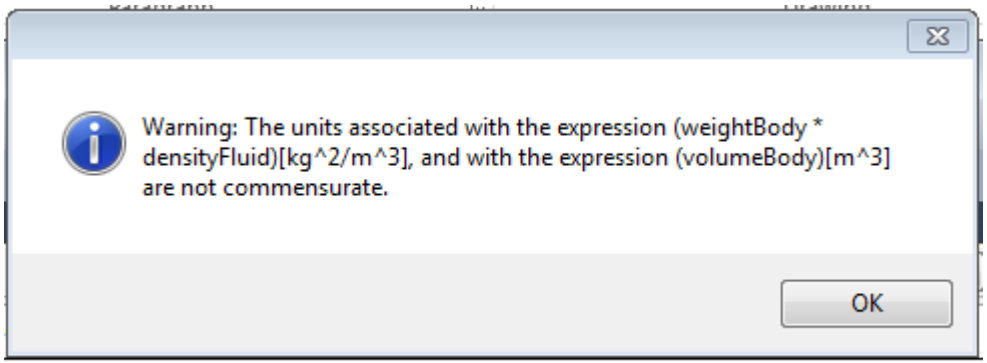
A. weightBody * densityFluid < volumeBody
B. weightBody < volumeBody * densityFluid

Which test is unit consistent?

1. A.
2. B.
3. Both.
4. Neither.
5. Does not matter.

Demo 3: AIMMS way of answering poll

- Demo with AIMMS or
- Click to open:



Unit casting

Escape clause:

- totalObjective := (w1 * obj1)[-] + (w2 * obj2)[-] ;

- When you feel you need it often; rethink.

Special case: Temperature

- Celsius and Kelvin
- Fahrenheit and Rankine

```
Quantity SI_Temperature {
    BaseUnit: K;
    Conversions: {
        C -> K : # -> # + 273.15,
        R -> K : # -> # * 0.5555555556,
        F -> K : # -> # * 0.5555555556 + 255.372222
```

- To increase the temperature of a body of 1 kg by 1 degree a certain amount of energy is needed:
- `unitEnergNeededIncrTemp` in [J / kg * K]

Remarks

- Logarithmic units, such dB (decibel) for sound, are not supported

- Units of measurement are not a **panacea**. They have to be used when necessary, otherwise an overkill of units might cause an application to be more complex than it should be. For instance, introducing units where integral value are expected such as `numberOfPlanes`.

Next Webinar

- Marcel Hunting
- October 15
- Introduction to Generated Mathematical Programs
  - Fast updates
  - Solution repository