Mass calculation

1 Total mass calculation

Now that the whole ship itself is in place, you are ready for some analysis functionality.

In entity `Mass calculation`, the mass and the centre of gravity of the whole system will be calculated. These values will depend on the **mass** and **centre of gravity** of both **light ship weight** and **cargo objects**. The total mass is calculated by the summation of all mass components (entities) one level below entity `Mass calculation`. The centre of gravity of a system of components is defined as the average of their positions, weighted by their masses.

- Include entities `Light Ship Weight` and `Mass Cargoes` as children of `Mass calculation`. Both are singular obligatory entities.
- Create the following parameters in the **Knowledge Browser**:

Parameter name	*Dimension*	*Determined by*	*Reference*	*In Class*
`Mass`	[t]	*USR: User or system/equation*	Mass	`Mass calculation`
`COGX`	[m]	*USR: User or system/equation*	Centre of gravity in X direction	`Dimensions`
`COGY`	[m]	*USR: User or system/equation*	Centre of gravity in Y direction	`Dimensions`
`COGZ`	[m]	*USR: User or system/equation*	Centre of gravity in Z direction	`Dimensions`
`MOMX`	[t*m]	*USR: User or system/equation*	Moment around X-axis	`Dimensions`
`MOMY`	[t*m]	*USR: User or system/equation*	Moment around Y-axis	`Dimensions`
`MOMZ`	[t*m]	*USR: User or system/equation*	Moment around Z-axis	`Dimensions`

- Include the following parameters in entity `Mass calculation`: `Mass`, `COGX`, `COGY`, `COGZ`, `MOMX`, `MOMY`, `MOMZ` and `QEntityRef`.
- Create the following relations in the **Knowledge Browser** and connect them to the pertaining parameters in entity `Mass calculation`.

`Mass = SUM(@QEntity, 1, @Mass) (`The result will be the summation of all parameters `Mass` of the entities one level below.)

`COGX = MOMX/Mass`

`COGY = MOMY/Mass`

`COGZ = MOMZ/Mass`

`MOMX = SUM(@QEntity, 1, @MOMX)`

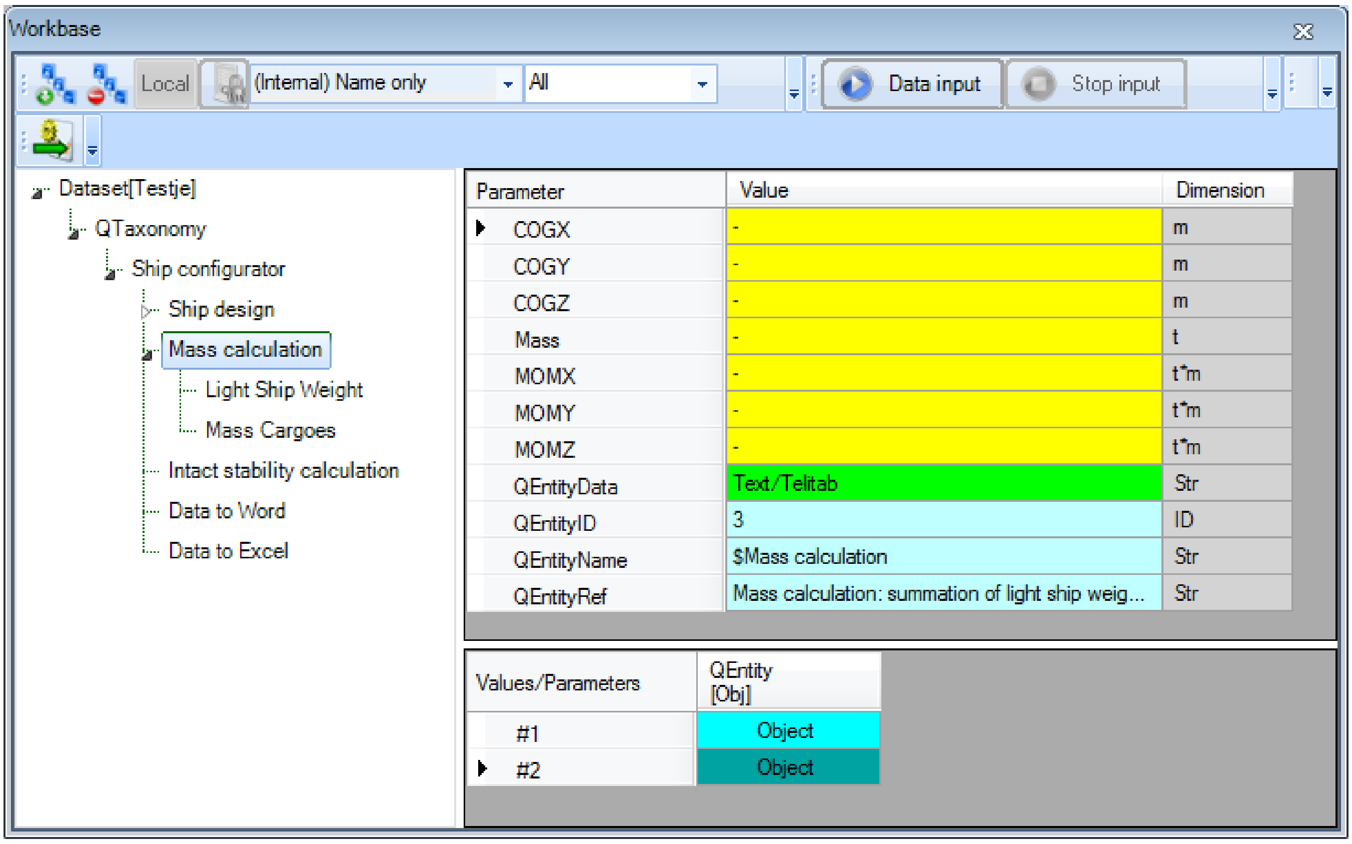
`MOMY = SUM(@QEntity, 1, @MOMY)`

`MOMZ = SUM(@QEntity, 1, @MOMZ)`

- To show computed values set attribute @SHOW on `QEntityData`.
- Assign the following text to `QEntityRef:` “Mass calculation: summation of light ship weight and total cargo weight”.

Please note, that you just created 'normal' relations and connected these to the parameters in the entity, because these are going to be used in several `Mass` entities.

Copyright © 2022, MARIN



- Dataset[Testje]
  - QTaxonomy
    - Ship configurator
      - Ship design
      - Mass calculation
        - Light Ship Weight
        - Mass Cargoes
      - Intact stability calculation
      - Data to Word
      - Data to Excel

Parameter	Value	Dimension
COGX	-	m
COGY	-	m
COGZ	-	m
Mass	-	t
MOMX	-	t*m
MOMY	-	t*m
MOMZ	-	t*m
QEntityData	Text/Telitab	Str
QEntityID	3	ID
QEntityName	$Mass calculation	Str
QEntityRef	Mass calculation: summation of light ship weig...	Str

Values/Parameters	QEntity [Obj]
#1	Object
#2	Object

2 Light Ship Weight

- Include the entities: `Mass Hull`, `Mass Decks` and `Mass Bulkheads` as children of `Light Ship Weight`. All three are singular obligatory.
- Include the following parameters in entity `Light Ship Weight`: `Mass`, `COGX`, `COGY`, `COGZ`, `MOMX`, `MOMY`, and `MOMZ`.
- Connect the same relations as above from the **Knowledge Browser** to the pertaining parameters in entity `Light Ship Weight`.
- To show computed values set attribute @SHOW on `QEntityData`.

Mass Hull

- Create the following parameter in the **Knowledge Browser**:

Parameter name	*Dimension*	*Determined by*	*Reference*	*In Class*
`Weight_volume_factor`	[t/m^3]	*USR: User or system/equation*	Weight factor per volume	`Mass calculation`

- Include the following parameters in entity `Mass Hull`: `Mass`, `COGX`, `COGY`, `COGZ`, `MOMX`, `MOMY`, `MOMZ`, `Volume` and `Weight_volume_factor`.
- Create the following relations in entity `Mass Hull` (here, 12 is the entity ID of `Main Dimensions`, change this for your case):

`Mass = Weight_volume_factor*Volume`

`COGX = (ENTITY#(12).Lpp)*0.5` (assume the centre of gravity of the hull in X direction half of the ship length)

`COGY = 0` (amidships)

`COGZ = (ENTITY#(12).Dm)*0.45` (assume the centre of gravity of the hull in Z direction at 45 % of the moulded depth)

`MOMX = COGX*Mass`

`MOMY = COGY*Mass`

`MOMZ = COGZ*Mass`

- To show computed values set attribute @SHOW on `QEntityData`.
- Make parameter `Volume` in entity `Mass Hull` use local properties ("Instantiate") and remove @MODIFY because you do not want to allow modification here.

- Create the following relation in entity `Mass Hull` (here, 13 is the entity ID of `Hydrostatics`, change this for your case):

```
Volume = ENTITY#(13).Volume
```

Mass Decks

- Include the singular obligatory entity `Decks` as child of entity `Mass Decks`.
- Include the following parameters in entity `Mass Decks`: `Mass`, `COGX`, `COGY`, `COGZ`, `MOMX`, `MOMY`, and `MOMZ`.
- Connect the following relations in the **Knowledge Browser** to the pertaining parameters in entity `Mass Decks` (This is again the same list as at the top of this page):

```
Mass = SUM(@QEntity, 1, @Mass)

COGX = MOMX/Mass

COGY = MOMY/Mass

COGZ = MOMZ/Mass

MOMX = SUM(@QEntity, 1, @MOMX)

MOMY = SUM(@QEntity, 1, @MOMY)

MOMZ = SUM(@QEntity, 1, @MOMZ)
```

- To show computed values set attribute @SHOW on `QEntityData`.

In entity `Decks`, below entity `Mass Decks`, you can present a table that contains a copy of parameters (`Name$` and `Area`) of all decks that are defined by the user in entity `Decks` below `Lay out`. In addition, the centre of gravity and mass of each deck can be calculated here.

- Include the following parameters in entity `Decks` below `Mass Decks`: `Name$`, `Nr`, `Area`, `Mass`, `COGX`, `COGY`, `COGZ`, `MOMX`, `MOMY`, `MOMZ` and `Weight_area_factor`.

- With exception of parameter `Nr`, put all parameters in the **table view** (localize (instantiate) parameter and provide attribute **@MULTVAL** in its **Data Slot**).

After the first parameter has been put in the table view with the attribute @MULTVAL, you can drag the other needed parameters directly to the table view from the Knowledge Browser.

The number of cases is equal to the number of defined decks by the user in the "**Lay out**" Entity. We are going to get this information.

- Create the following relation in entity `Mass Decks->Decks` (here, 10 is the entity ID of `Layout->Decks`)

```
Nr = ENTITY#(10).Nr
```

The first column/case of the table should contain data (`Name$` and `Area`) from the first defined Deck in entity `Deck`**.** The second column contains data from the second defined deck, etc.

- This can be done by the following relations in entity `Mass Decks->Decks` (here, 16 is the entity ID of `Layout->Decks->Deck`)

```
Name$ = ENTITY#(16, ORCA(1)).Name$

Area = ENTITY#(16, ORCA(1)).Area
```

When you want refer to a multiple entity, you also have to indicate the QEntityIndex. Quaestor automatically provides an index value in the Quaestor parameter `QEntityIndex` for each multiple entity. So ENTITY#(xx, 3) refers to the third defined entity `Deck`. The function **ORCA(1)** returns the current case number which is now being executed. So for the second column/case in a table, the value of ORCA(1) = 2. When we combine the index with the ORCA() function, like in the `Area` relation above, the second column of the current table will refer to the area from the second defined deck, etc.

- Add the following relations in entity `Mass Decks->Decks` (here, 16 is the entity ID of `Layout->Decks->Deck`):

```
COGX = ENTITY#(16, ORCA(1.X_front) + (ENTITY#(16, ORCA(1)).X_aft))/2

COGY = 0

COGZ = ENTITY#(16, ORCA(1)).Z

MOMX = COGX*Mass

MOMY = COGY*Mass

MOMZ = COGZ*Mass

Mass = Weight_area_factor*Area
```

- To show computed values set attribute @SHOW on `QEntityData`.

As already discussed in the `Deck` entity, the `Weight_area_factor` is a bit special. What you want is, that this property is connected to the original value in `Deck` (of which the value is hidden) but, when modified by the user, the modified value should be used in both the present `Mass Decks->Decks` entity AND the original entity `Decks->Deck`. This is done by means of the **@SAVEINSOURCE** attribute.

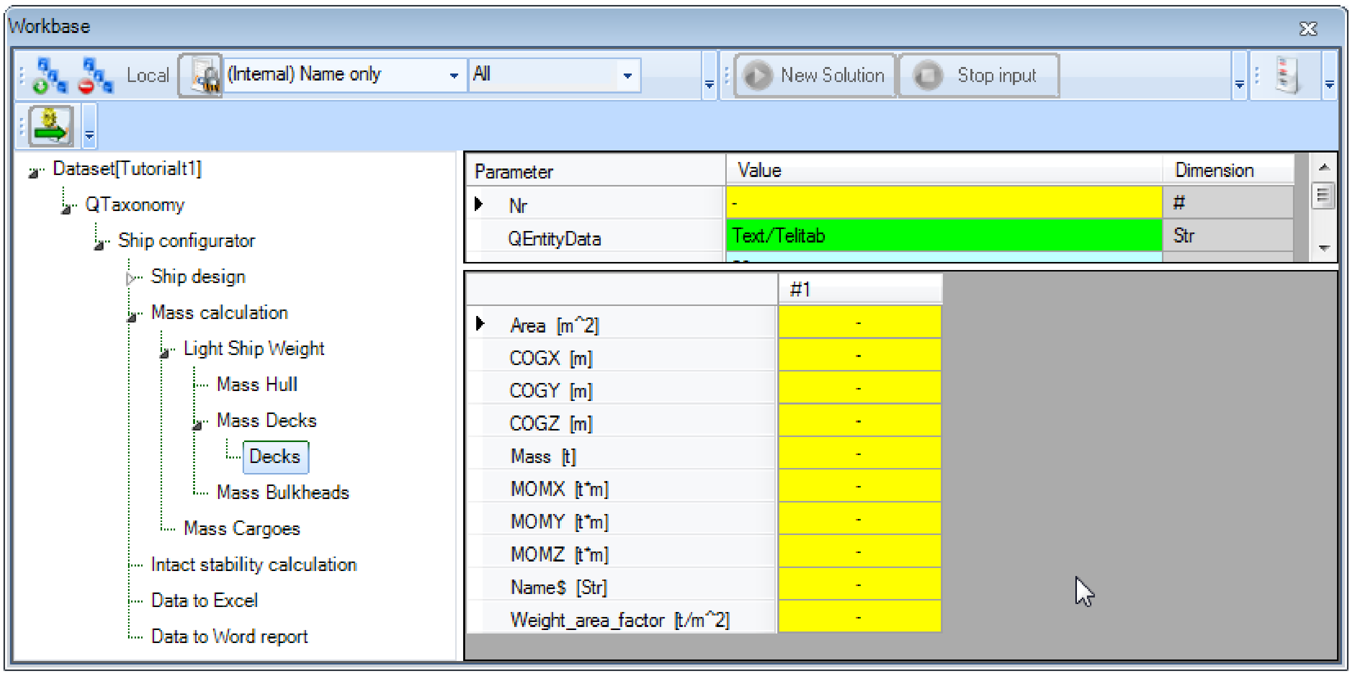
- Create the following relation:

```
Weight_area_factor = ENTITY#(16, ORCA(1)).Weight_area_factor
```

- And add the following attributes to the localized `Weight_area_factor` parameter:

@MULTVAL
@MODIFY
@SAVEINSOURCE

When a user adds or removes a deck, or changes values of an existing defined deck, these adaptations will automatically propagated to the `Mass Decks` entity.



(Internal) Name only | All

- Dataset[Tutorialt1]
  - QTaxonomy
    - Ship configurator
      - Ship design
      - Mass calculation
        - Light Ship Weight
          - Mass Hull
          - Mass Decks
            - Decks
          - Mass Bulkheads
        - Mass Cargoes
      - Intact stability calculation
      - Data to Excel
      - Data to Word report

Parameter	Value	Dimension
Nr	-	#
QEntityData	Text/Telitab	Str

	#1
Area [m^2]	-
COGX [m]	-
COGY [m]	-
COGZ [m]	-
Mass [t]	-
MOMX [t*m]	-
MOMY [t*m]	-
MOMZ [t*m]	-
Name$ [Str]	-
Weight_area_factor [t/m^2]	-

Mass Bulkheads

- Include the singular obligatory entity `Bulkheads` as child of entity `Mass Bulkheads`.
- Include the following parameters in entity `Mass Bulkheads`: `Mass`, `COGX`, `COGY`, `COGZ`, `MOMX`, `MOMY`, and `MOMZ`.
- Connect the following relations in the **Knowledge Browser** to the pertaining parameters in entity `Mass Bulkheads` (This is again the same list as at the top of this page):

```
Mass = SUM(@QEntity, 1, @Mass)

COGX = MOMX/Mass

COGY = MOMY/Mass

COGZ = MOMZ/Mass

MOMX = SUM(@QEntity, 1, @MOMX)

MOMY = SUM(@QEntity, 1, @MOMY)

MOMZ = SUM(@QEntity, 1, @MOMZ)
```

- To show computed values set attribute @SHOW on `QEntityData`.
- Include the following parameters in entity `Bulkheads` below `Mass Bulkheads`: `Name$`, `Nr`, `Area`, `Mass`, `COGX`, `COGY`, `COGZ`, `MOMX`, `MOMY`, `MOMZ` and `Weight_area_factor`.
- With exception of parameter `Nr`, put all parameters in the **table view** (localize (instantiate) parameter and provide attribute **@MULTVAL** in its **Data Slot**).

- Make the same kind of relations in `Mass Bulheads->Bulkheads` as in `Mass Decks->Decks` above (here, 17 refers to `Layout->Bulkheads->Bulkheads`):

```
Nr = ENTITY#(17).Nr

Area = ENTITY#(17).Area.ORCA(1)

Name$ = ENTITY#(17).Name$.ORCA(1)
```
TODO: dit levert een foutmelding in Quaestor

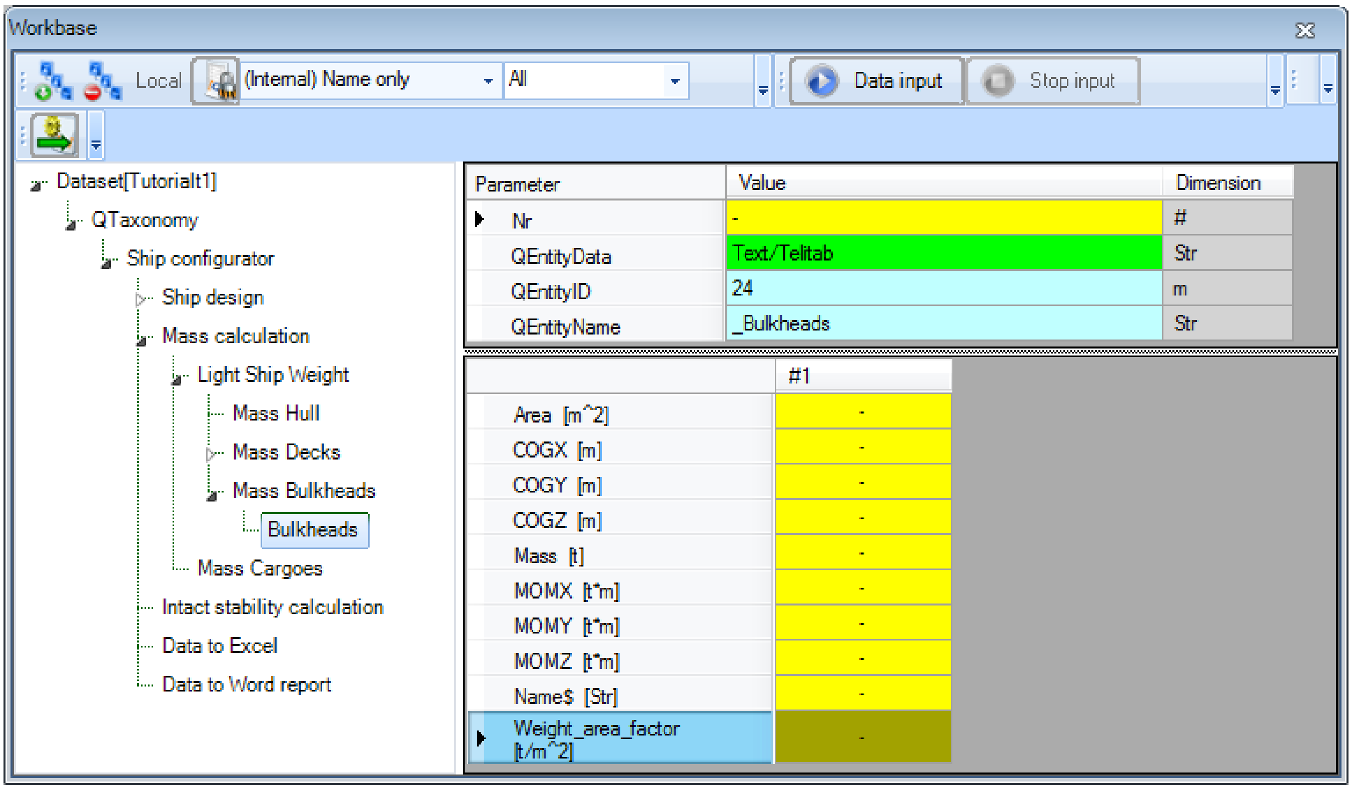
```
COGX = ENTITY#(17).X.ORCA(1)

COGY = 0

COGZ = (ENTITY#(17).Z_bottom.ORCA(1) + ENTITY#(17).Z_top.ORCA(1))/2

MOMX = COGX*Mass

MOMY = COGY*Mass

MOMZ = COGZ*Mass

Mass = Weight_area_factor*Area

Weight_area_factor = ENTITY#(17).Weight_area_factor.ORCA(1)
```

- And add the following attributes to the localized `Weight_area_factor` parameter:

@MULTVAL
@MODIFY
@SAVEINSOURCE

- To show computed values during a dialogue write "@SHOW" behind "**QEntityData**"

Please note the difference with the previous section, in which a table was created referring to values within multiple entities. For the bulkheads we have created one table and refer to values within the table of another singular Entity!



Parameter	Value	Dimension
Nr	-	#
QEntityData	Text/Telitab	Str
QEntityID	24	m
QEntityName	_Bulkheads	Str

	#1
Area [m^2]	-
COGX [m]	-
COGY [m]	-
COGZ [m]	-
Mass [t]	-
MOMX [t*m]	-
MOMY [t*m]	-
MOMZ [t*m]	-
Name$ [Str]	-
Weight_area_factor [t/m^2]	-

3 Mass Cargoes

- Include the singular obligatory entity `Cargoes` as child of entity `Mass Cargoes`.
- Include the following parameters in entity `Mass Cargoes`: `Mass`, `COGX`, `COGY`, `COGZ`, `MOMX`, `MOMY`, `MOMZ and QEntityRef`.

- Connect the same relations as mentioned at the top of the page from the **Knowledge Browser** to the pertaining parameters in entity `Mass Cargoes`.
- To show computed values set attribute @SHOW on `QEntityData`.
- Assign the following value to `QEntityRef`: “Cargo mass calculation: summation of all cargo components”.
- Include the following parameters in entity `Cargoes` below `Mass Cargoes`: `Name$`, `Nr`, `Mass`, `COGX`, `COGY`, `COGZ`, `MOMX`, `MOMY`, `MOMZ`, `QEntityDoc` and `QEntityRef`.
- With exception of parameter `Nr`, put all parameters in the **table view** (localize (instantiate) parameter and provide attribute **@MULTVAL** in its **Data Slot**).

In entity `Cargoes` you will enable the user to create a table with a number of cargo objects. For each object, the user has to provide a name, COG and mass.

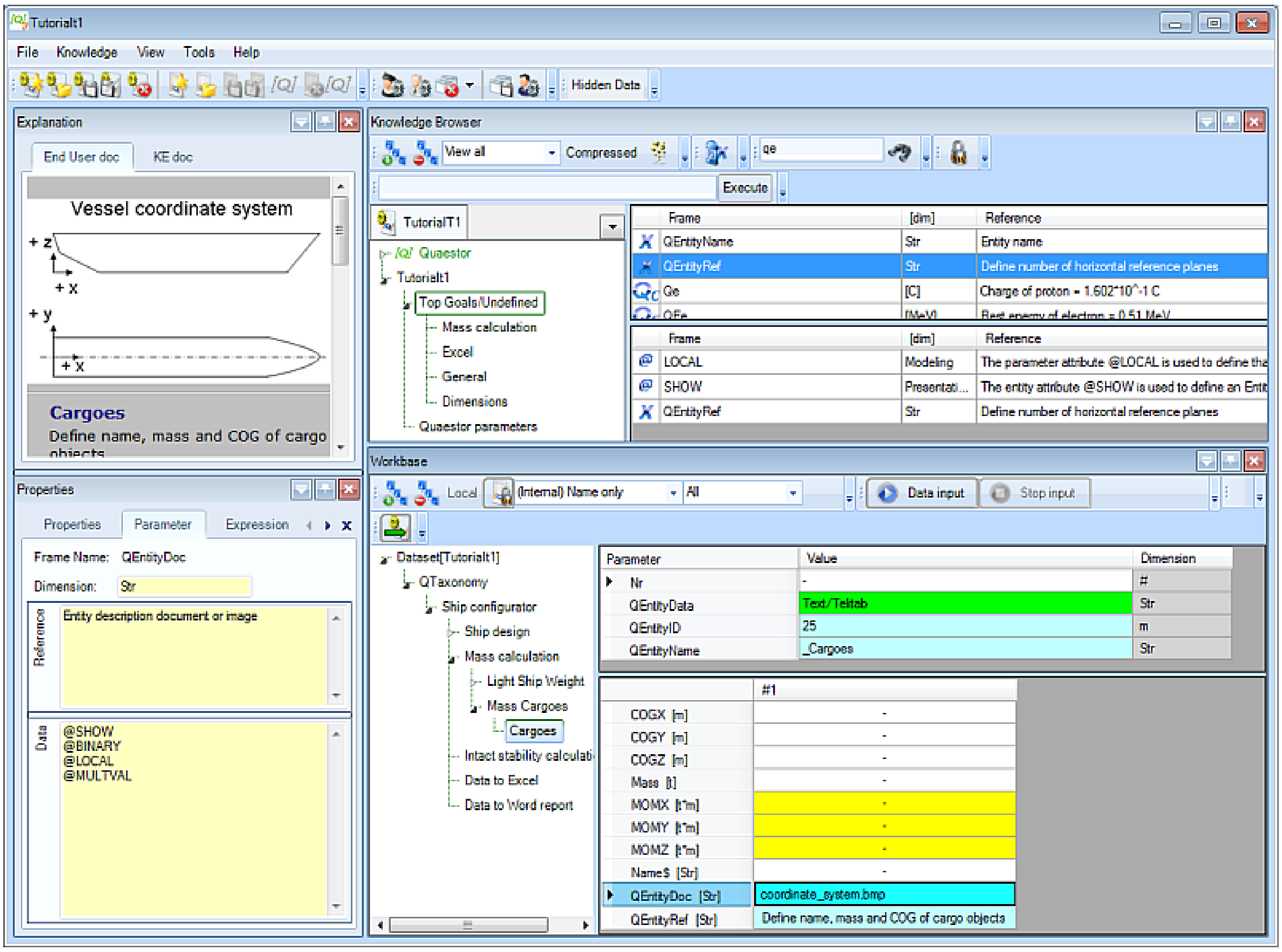
- Create the following relations:

```
MOMX = COGX*Mass

MOMY = COGY*Mass

MOMZ = COGZ*Mass
```

- To show computed values set attribute @SHOW on `QEntityData`.
- Assign the following value to `QEntityRef`: “Define name, mass and COG of cargo objects”

As explanation to the user, you can include a schematic picture that depicts the used coordinate system within this configurator.

- *Right-click* on `QEntityDoc` in entity `Cargoes` and select *Taxonomy->Include Binary Data* or press *Ctrl+B*. Now you can browse to the file you want to include: “coordinate_system.bmp”.



Cargoes
Define name, mass and COG of cargo objects

Frame	[dim]	Reference
QEntityName	Str	Entity name
QEntityRef	Str	Define number of horizontal reference planes
Qe	[C]	Charge of proton = 1.602*10^-1 C
QEe	[MeV]	Rest energy of electron = 0.51 MeV

Frame	[dim]	Reference
LOCAL	Modeling	The parameter attribute @LOCAL is used to define the
SHOW	Presentati...	The entity attribute @SHOW is used to define an Enti
QEntityRef	Str	Define number of horizontal reference planes

Frame Name: QEntityDoc
Dimension: Str
Reference: Entity description document or image
Data: @SHOW @BINARY @LOCAL @MULTVAL

Parameter	Value	Dimension
Nr	-	#
QEntityData	Text/Teltab	Str
QEntityID	25	m
QEntityName	_Cargoes	Str

	#1
COGX [m]	-
COGY [m]	-
COGZ [m]	-
Mass [t]	-
MOMX [t*m]	-
MOMY [t*m]	-
MOMZ [t*m]	-
Name$ [Str]	-
QEntityDoc [Str]	coordinate_system.bmp
QEntityRef [Str]	Define name, mass and COG of cargo objects

Back to content | << Previous | Next >>