Using SysML elements in UAF plugin

SysML elements can be used to extend UAF provided set of concepts. SysML toolbars in UAF diagrams are collapsed by default.

For better understanding how to use SysML elements in UAF plugin models two case studies are described:

- Case Study 1. The Requirement usage in OV-2 diagram
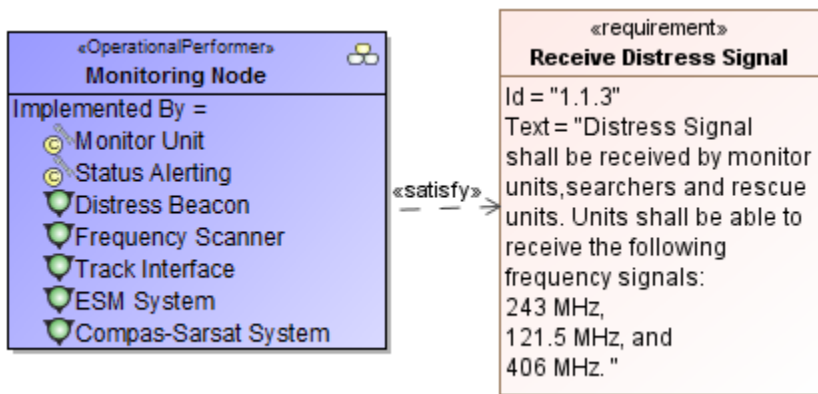- Case Study 2. Describing a full-time equivalent (FTE) calculations in the UAF model

Case Study 1. The Requirement usage in OV-2 diagram

We will use the Requirement element, which is a SysML element, in an OV-2 diagram.

To use the Requirement in an OV-2 diagram

1. Create an OV-2 diagram.
2. Create an Operational Performer in the diagram pane.
3. Expand the SysML Requirements Diagram toolbar and select Requirement.
4. Create it in the diagram pane.
5. Create the Satisfy relationship between the Operational Performer and the Requirement.

As you can see in the following figure, the architecture element satisfies the Requirement.

Case Study 2. Describing a full-time equivalent (FTE) calculations in the UAF model
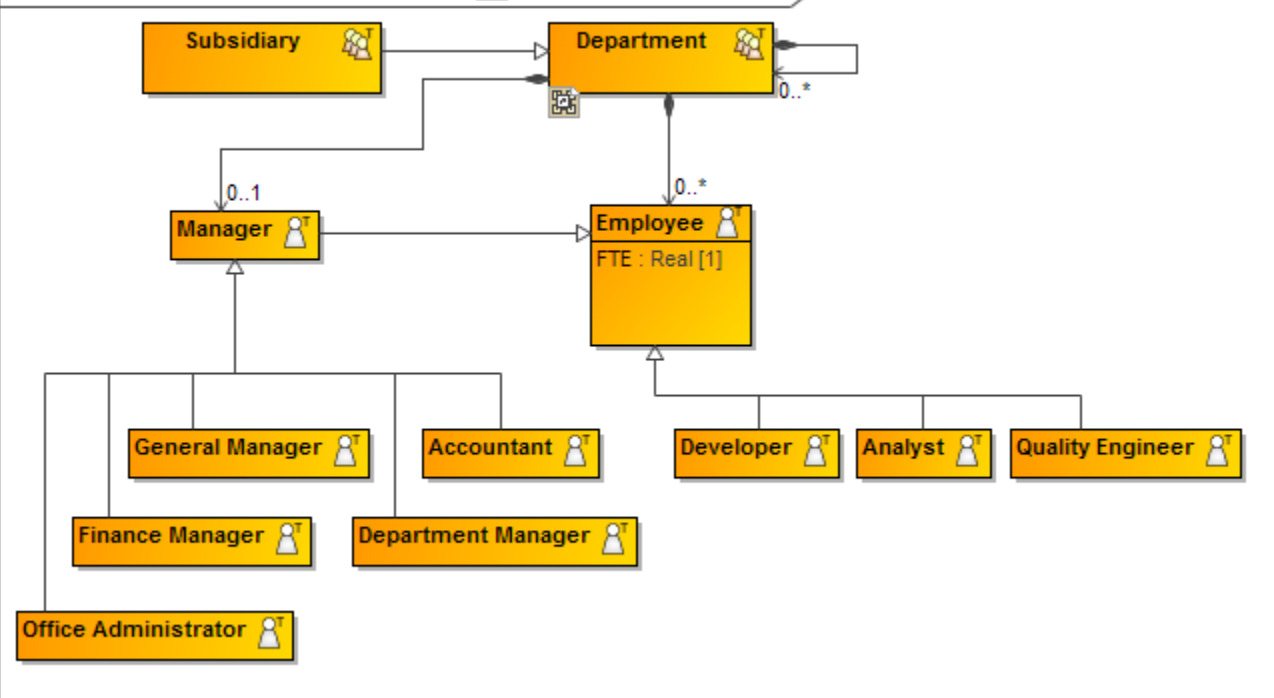
The following example describes the actual situation, how you can use the SysML elements in the UAF model.

> ⓘ The case study uses the sample project Counting FTEs (requires CST).mdzip, which can be found in <MagicDraw installation folder>\samples\UAF.

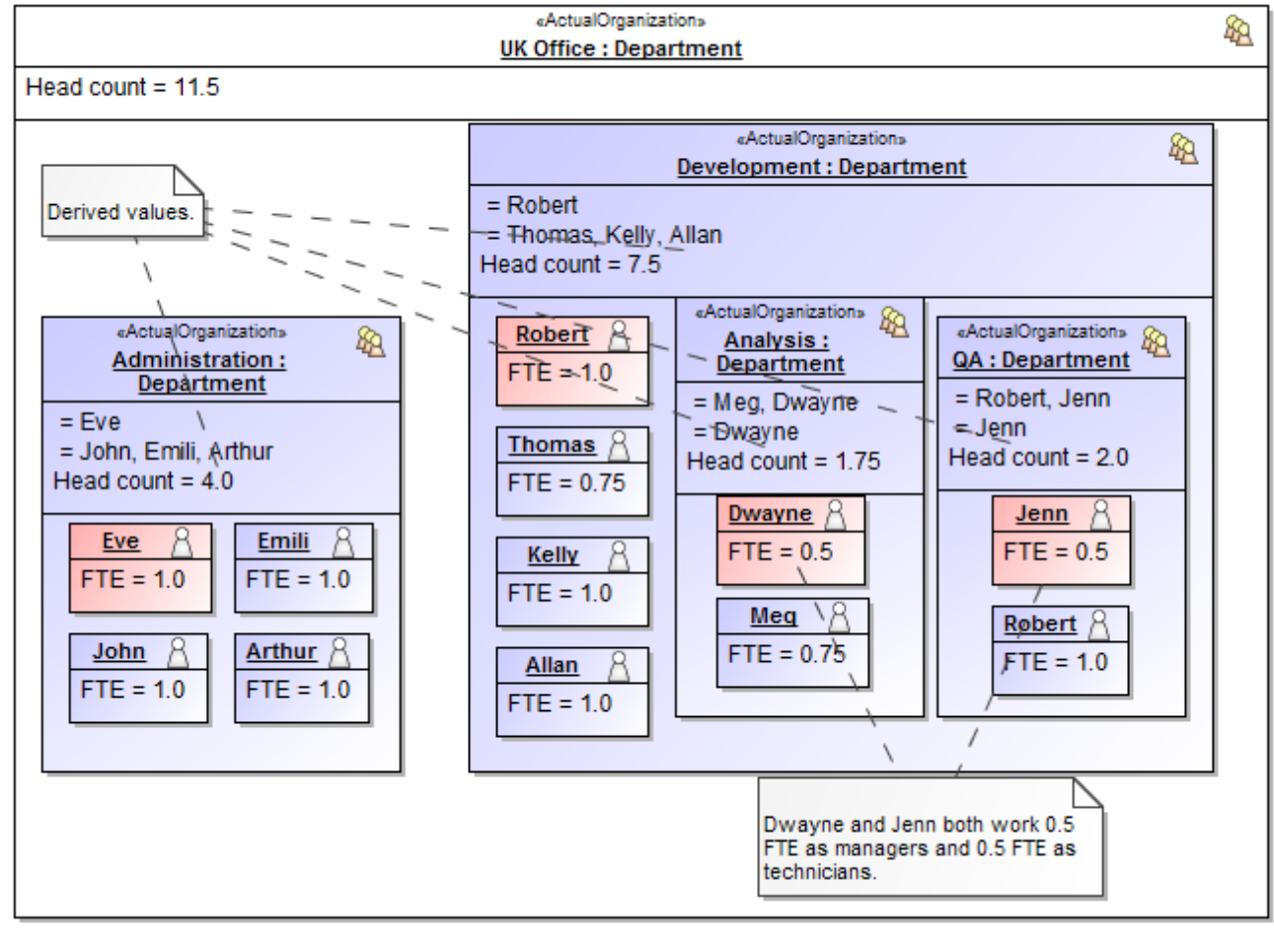Let us say, we need to describe a full-time equivalent (FTE) calculations in the UAF model.

First, we need to create the typical organizational structure using OV-4 Organizational Relationship Chart diagram. The typical organizational structure defines types of departments and roles and how they are associated together in the modeled enterprise. The Department can contain any number of Employees and any number of sub Departments.

OV-4 Organizational Relationships Chart [▦ Typical Organizational Chart]

- Subsidiary
- Department
  - 0..*
  - Manager
    - 0..1
    - General Manager
    - Finance Manager
    - Office Administrator
    - Accountant
    - Department Manager
  - Employee
    - FTE : Real [1]
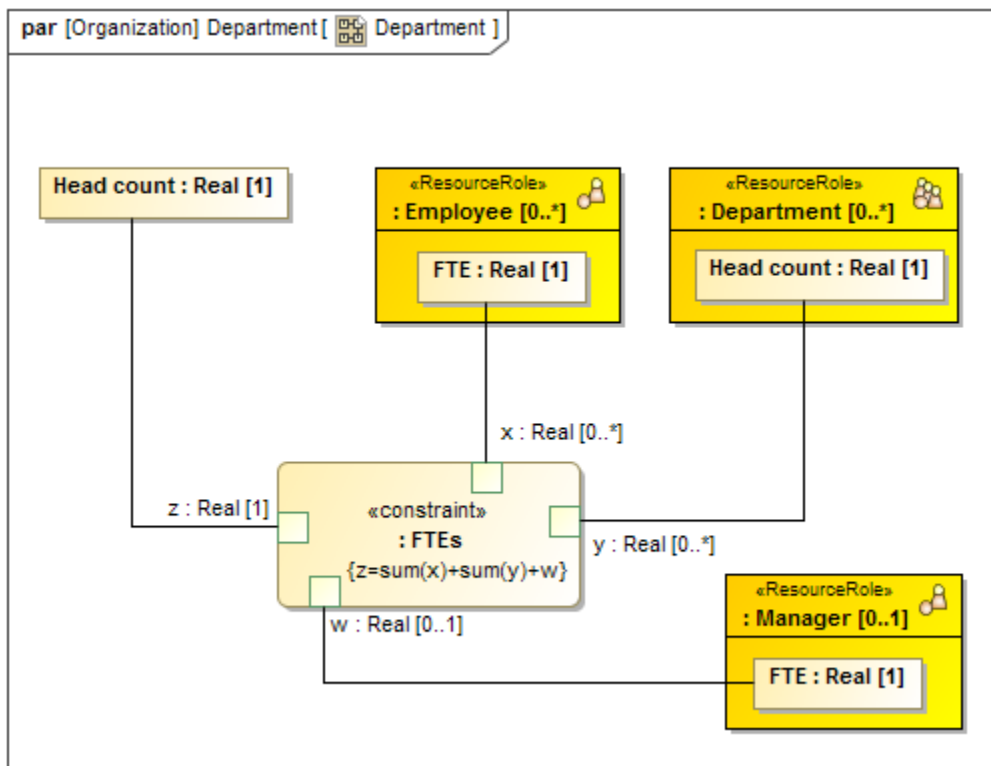   - 0..*
    - Developer
    - Analyst
    - Quality Engineer

Secondly, we generate an actual organizational chart based on the typical organizational chart (see Instantiating structures). An actual organizational chart shows concrete departments and residing employees.

Each employee works a certain number of FTEs.

OV-4 Organizational Relationships Chart [🖳 Instance of the UK Office]

«ActualOrganization»
UK Office : Department

Head count = 11.5

«ActualOrganization»
Development : Department

= Robert
= Thomas, Kelly, Allan
Head count = 7.5

Derived values.

«ActualOrganization»
**Administration :
Department**

= Eve
= John, Emili, Arthur
Head count = 4.0

Eve 👤	**Emili** 👤
FTE = 1.0	FTE = 1.0

John 👤	**Arthur** 👤
FTE = 1.0	FTE = 1.0

Robert 👤
FTE = 1.0

Thomas 👤
FTE = 0.75

Kelly 👤
FTE = 1.0

Allan 👤
FTE = 1.0

«ActualOrganization»
**Analysis :
Department**

= Meg, Dwayne
= Dwayne
Head count = 1.75

Dwayne 👤
FTE = 0.5

Meg 👤
FTE = 0.75

«ActualOrganization»
QA : Department

= Robert, Jenn
= Jenn
Head count = 2.0

Jenn 👤
FTE = 0.5

Robert 👤
FTE = 1.0

Dwayne and Jenn both work 0.5
FTE as managers and 0.5 FTE as
technicians.

Thirdly, we can use the SysML Parametric Diagram to describe the recursive FTE's calculation for the selected department and each residing sub-department.

par [Organization] Department [🔲 Department]

Head count : Real [1]

«ResourceRole»
: Employee [0..*]

FTE : Real [1]

«ResourceRole»
: Department [0..*]

Head count : Real [1]

x : Real [0..*]

z : Real [1]

«constraint»
: FTEs
{z=sum(x)+sum(y)+w}

y : Real [0..*]

«ResourceRole»
: Manager [0..1]

w : Real [0..1]

FTE : Real [1]

ⓘ You can execute the described calculations using Cameo Simulation Toolkit.