<config>  
</config>

<**context** model= ‘**$PROJECT\_LOC**\VintageCarParts.factory’ element=’{0}’ searchMetamodels='true' importedBundles=’gmf;sirius’ />

<**gendoc** id=’title’>

[self.oclAsType(FactorySpecification).businessTypeName /] Owned By

[self.oclAsType(FactorySpecification).label /]

Table of content

[1. Summary 3](#_Toc455334769)

**Table of Figures**

[Figure 1 : 3](#_Toc420585209)

</gendoc><drop/>

<**gendoc** id=’summary’><drop/>

# Summary

[if self.oclAsType(FactorySpecification).ownedFactories->notEmpty()]<drop/>

Table 1 lists the [self.oclAsType(FactorySpecification).businessTypeName.toLower()/] owned by [self.oclAsType(FactorySpecification).label/]:

<table><drop/>

**Table 1 : The** list of [self.oclAsType(FactorySpecification).businessTypeName.toLower()/] owned by [self.oclAsType(FactorySpecification).label/]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Address | Dimensions (meters) | Employees | Products Made |

[for (factory : factory::Factory | self.oclAsType(FactorySpecification).ownedFactories->sortedBy( label ) )]<drop/>

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **[factory.label/]** | [factory.streetNumber/] [factory.street/]  [factory.postalCode/] [factory.city/]  [factory.country/] | [factory.width/] X [factory.length/] | [for ( employee : factory::Employee | factory.employees->sortedBy( label ) )]<drop/>  [employee.label/]  [/for]<drop/> | [for ( productPart : factory::ProductPart | factory.createdProductParts->sortedBy( label ) )]<drop/>  [productPart.label/]  [/for]<drop/> |

[/for]<drop/>

</table><drop/>

Each factory is described in details in the following sections.

[else] <drop/>

No factories.

[/if]<drop/>

</gendoc><drop/>

<**gendoc** id=’factoryDescription’><drop/>

[for (factory : factory::Factory | self.oclAsType(FactorySpecification).ownedFactories->sortedBy( label ) )]<drop/>

# [factory.label/]

[factory.description/] The diagram of Figure 1 shows an overview of the factory and its working employees.

[for (di: notation::Diagram | factory.getSiriusDiagrams('representations.aird', factory.label))]<drop/>

<image object='[di.getDiagramExt('jpg')/]' maxW='true' maxH=’true’> <drop/>

Figure : Overview of [factory.label/]

</image><drop/>

[/for]<drop/>

[for (productPart : factory::ProductPart | factory.createdProductParts->sortedBy( label ) )]<drop/>

## [productPart.label/]

### Overview

An exploded view of a [productPart.label/] for the [productPart.supportedProductVersions->first().eContainer().oclAsType(factory::Product).label/] is shown in Figure 2. It is compatible with the [if productPart.supportedProductVersions->first() = productPart.supportedProductVersions->last()][productPart.supportedProductVersions->first().label/] model variant[else][for (version : factory::ProductVersion | productPart.supportedProductVersions->sortedBy( label ) )][if version = productPart.supportedProductVersions->sortedBy( label )->last()]and [version.label/][else][version.label/], [/if][/for] model variants[/if].

<image filePath=’[’E:/Dokumente/Studium/Master/MDSD/mdsd-factory-project/notation/de.mdelab.languages.factory.instance/documentation/images/’.concat(productPart.label).concat(‘.png’)/]’ maxW='true' maxH=’true’> <drop/>

Figure 2 : Exploded view of a [productPart.label/] for the [productPart.supportedProductVersions->first().eContainer().oclAsType(factory::Product).label/]

</image><drop/>

### Production Schemata

[if (not productPart.constructionProcesses->isEmpty())]<drop/>

Figure 3 shows the production schemata [productPart.constructionProcesses->first().label/] for [productPart.label/]s produced at [factory.label/].

[for (di: notation::Diagram | productPart.constructionProcesses->first().getSiriusDiagrams('representations.aird', productPart.constructionProcesses->first().label))]<drop/>

<image object='[di.getDiagramExt('jpg')/]' maxW='true' maxH=’true’> <drop/>

Figure : The production schemata for [productPart.label/] produced at [factory.label/]

</image><drop/>

[/for]<drop/>

[if (not productPart.constructionProcesses->first().description.oclIsUndefined())]<drop/>

[productPart.constructionProcesses->first().description/]

[/if]<drop/>

[else]<drop/>

None

[/if]<drop/>

[/for]<drop/>

[/for]<drop/>

</gendoc><drop/>

<gendoc id=’carModels’><drop/>

# Car Models

This section presents the car models for which parts are produced by the factories owned by [self.oclAsType(FactorySpecification).label/].

[for (car : factory::Product | self.oclAsType(FactorySpecification).ownedProducts->sortedBy( label ) )]<drop/>

## [car.label/]

Figure 1 shows a car of the [car.label/] series that was produced between [car. productionStartYear/] and [car.productionEndYear/] in [car.ownedProductVersions->size()/] different versions:

[for (productVersion : factory::ProductVersion | car.ownedProductVersions->sortedBy( label ) )]<drop/>

* [productVersion.label/][if (not productVersion.description.oclIsUndefined())] ([productVersion.description/])[/if]

[/for]<drop/>

<image filePath=’[’E:/Dokumente/Studium/Master/MDSD/mdsd-factory-project/notation/de.mdelab.languages.factory.instance/documentation/images/’.concat(car.label).concat(‘.png’)/]’ maxW='true' maxH=’true’> <drop/>

Figure  : A car of the [car.label/] series

</image><drop/>

[/for]<drop/>

</gendoc><drop/>