<config><drop/>

<param key=’**modelPath**’ value=’${project\_loc}/Models/alarm\_clock\_radio.uml’/>

<param key=’**systemName**’ value=’Alarm clock radio’/>

<param key=’**requirementsPackage**’ value=’SysMLmodel/requirements’/>

<param key=’**ucPackage**’ value=’SysMLmodel/context’/>

<param key=’**blocksPackage**’ value=’SysMLmodel/structure’/>

**<output path**=’${project\_loc}/Documents/Generated/Alarm\_clock\_radio.docx’/>

</config><drop/>

<**context** model=’${modelPath}’ importedBundles=’gmf;papyrus’ searchMetamodels=’true’/>

<**gendoc** id=’title’>

[gGet(‘systemName’)/]

**</gendoc>**

# System requirements

The following table displays the requirements for the system :

<**context** element=’${requirementsPackage}’ /><drop/>

<**gendoc** id=’requirements’><drop/>

<table><drop/>

|  |  |  |
| --- | --- | --- |
| ID | Abstract | Description |

[for (req : Requirement|Requirement.allInstances()->sortedBy(id))]

|  |  |  |
| --- | --- | --- |
| [req.id/] | [req.base\_Class.name/] | [req.text/] |

[/for]

</table><drop/>

</gendoc><drop/>

# Context

<context element=’${ucPackage}’ /><drop/>

**<gendoc** id=’context’><drop/>

[for (diag: Diagram|self.getPapyrusDiagrams())]<drop/>

The following use case diagram illustrates the context of the system ***[diag.name/] :***

<image object='[diag.getDiagram()/]' maxW=’true’><drop/>

</image><drop/>

The following actors are displayed:

[for (a:Actor | diag.getElementsInDiagram()->filter(Actor)->sortedBy(name))]<drop/>

* [a.name/]

<dropEmpty>*[a.getDocumentation()/]*</dropEmpty>

[/for]<drop/>

The following use cases are displayed:

[for (uc : UseCase | diag.getElementsInDiagram()->filter(UseCase))]<drop/>

* [uc.name/]

<dropEmpty>*[uc.getDocumentation()/]</*dropEmpty>

[if(uc.classifierBehavior.getPapyrusDiagrams()->notEmpty())]<drop/>

### Behavior diagrams

[/if]<drop/>

[for (diag2: Diagram| uc.classifierBehavior.getPapyrusDiagrams())]<drop/>

This use case is described by the following behavior diagram ***[diag2.name /] :***

<image object='[diag2.getDiagram()/]' maxW=’true’><drop/>

</image><drop/>

[/for]<drop/> -- end Behavior diagrams

[/for]<drop/> -- end of UC loop

[/for]<drop/> -- End of diagram loop

</gendoc><drop/>

# System structure breakdown

<**context** element=’${blocksPackage}’ /><drop/>

<**gendoc** id=’structure’>

## Block definition diagram

[for (diag: Diagram|self.getPapyrusDiagrams())]<drop/>

***[diag.name/]***

<image object='[diag.getDiagram()/]' maxW=’true’><drop/>

</image>

[/for] <drop/>

[for (b:Block |self.ownedElement->filter(Class).getStereotypeApplications()->select(oclIsKindOf(Block)).oclAsType(Block)->sortedBy(base\_Class.name))]<drop/>

## Block [b.base\_Class.name/]

<dropEmpty>*[b.getDocumentation()/]</*dropEmpty>

[if(b.base\_Class.ownedAttribute->filter(Property)->reject(oclIsKindOf(Port))->notEmpty())]<drop/>

### Block properties

[/if]<drop/>

[for (p:uml::Property |b.base\_Class.ownedAttribute->filter(Property)->reject(oclIsKindOf(Port))->sortedBy(name))]<drop/>

* [p.name/] [if (not (p.type.oclIsUndefined()))]: *[p.type.name/]*[/if]

<dropEmpty>*[p.getDocumentation()/]</*dropEmpty>

[/for]<drop/>

[if(b.base\_Class.ownedAttribute->filter(Property)->reject(oclIsKindOf(Port))->select(aggregation=’composite’)->notEmpty())]<drop/>

### Block parts

[/if]<drop/>

[for (p:uml::Property |b.base\_Class.ownedAttribute->filter(Property)->reject(oclIsKindOf(Port))->select(aggregation=’composite’)->sortedBy(name))]<drop/>

* [p.name/] [if (not (p.type.oclIsUndefined()))]: *[p.type.name/]*[/if]

<dropEmpty>*[p.getDocumentation()/]</*dropEmpty>

[/for]<drop/>

[if(b.base\_Class.ownedAttribute->filter(Port)->notEmpty())]<drop/>

### Block ports

[/if]<drop/>

[for (p: Port |b.base\_Class.ownedAttribute->filter(Port)->sortedBy(name)) ]<drop/>

* **[p.getStereotypeApplications()->select(oclIsTypeOf(FlowPort)).oclAsType(FlowPort).direction/]** [p.name/] [if (not (p.type.oclIsUndefined()))]: *[p.type.name/]*[/if]

<dropEmpty>*[p.getDocumentation()/]</*dropEmpty>

[/for]<drop/>

[if(b.base\_Class.getPapyrusDiagrams()->notEmpty())]<drop/>

### Internal block diagram

[/if]<drop/>

[for (diag: Diagram|b.base\_Class.getPapyrusDiagrams())]<drop/>

This block is described by the following internal block diagram ***[diag.name/] :***

<image object='[diag.getDiagram()/]' maxW=’true’>

</image>

[/for] <drop/>

[if(b.base\_Class.ownedBehavior.getPapyrusDiagrams()->notEmpty())]<drop/>

### Behavior diagrams

[/if]<drop/>

[for (diag: Diagram| b.base\_Class.ownedBehavior.getPapyrusDiagrams())]<drop/>

This block is described by the following behavior diagram ***[diag.name/] :***

<image object='[diag.getDiagram()/]' maxW=’true’>

</image>

[/for] <drop/>

[/for]<drop/> -- end block

</gendoc><drop/>