Architecture File of {m:self.name}

Document generated automatically by M2Doc from Capella model of the system

# Table of Contents

{m:'newTableOfContent'.asPagination()}

# Descriptions of Logical Components

{m:for lc | self.eAllContents (la::LogicalComponent)}

## Logical Component : {m:lc.name}

{m:if lc.description.trim().size() <> 0}

{m:lc.description.trim().fromHTMLBodyString().replaceLink(lc)}

{m:else}

*No description*

{m:endif}

{m:if lc.isRepresentationDescriptionName('Logical Component Breakdown')}

{m:lc.asImageByRepresentationDescriptionName('Logical Component Breakdown')->first().setWidth(400)}

{m:endif}

{m:if lc.realizedSystemComponents->size() <> 0}

### Realizing system components:

{m:for systComp| lc.realizedSystemComponents.eContainerOrSelf(capellacore::NamedElement)}

* {m:systComp.name}

{m:endfor}

{m:endif}

{m:if lc.realizingPhysicalComponents->size() <> 0}

### Realizing physical components:

{m:for physComp| lc. realizingPhysicalComponents.eContainerOrSelf(capellacore::NamedElement)}

* {m:physComp.name}

{m:endfor}

{m:endif}

{m:if lc.allocatedFunctions->size() <> 0}

### Allocated Functions

|  |  |
| --- | --- |
| **Function name** | **Description** |

{m:for function | lc.allocatedFunctions}

|  |  |
| --- | --- |
| {m:function.name} | {m:function.description.trim().fromHTMLBodyString().replaceLink(function)} |

{m:endfor}

{m:endif}

{m:if lc.ownedFeatures->filter (fa::ComponentPort).componentExchanges->size() <> 0}

### Component Exchanges

{m:for ce | lc.ownedFeatures->filter (fa::ComponentPort).componentExchanges}

|  |  |
| --- | --- |
| **Name** | {m:ce.name} |
| **Direction** | {m:ce.getCeDirection(lc)} |
| **Destination Component** | {m:ce.getDestinationComponent(lc)} |

{m:if ce.description.trim().size() <> 0}

{m:ce.description.trim().fromHTMLBodyString().replaceLink(ce)}

{m:else}

*No description*

{m:endif}

{m:endfor}

{m:endif}

{m:for interface | lc.requiredInterfaces}

### Required Interfaces {m:interface.name}

{m: interface.description.trim().fromHTMLBodyString().replaceLink(interface)}

{m:endfor}

{m:for interface | lc.providedInterfaces}

### Provided Interfaces {m:interface.name}

{m: interface.description.trim().fromHTMLBodyString().replaceLink(interface)}

{m:endfor}

{m:endfor}

# Descriptions of Logical Functions

{m:for f | self.eAllContents (la::LogicalFunction)}

## Function : {m:f.name}

Description:

{m:if f.description.trim().size() <> 0}

{m:f.description.trim().fromHTMLBodyString().replaceLink(f)}

{m:else}

*No description*

{m:endif}

{m:if f.isRepresentationDescriptionName('Logical Data Flow Blank')}

{m:f.asImageByRepresentationDescriptionName('Logical Data Flow Blank').setWidth(400)}

{m:endif}

Parent / Children functions:

|  |  |  |
| --- | --- | --- |
| **Parent function** | **Current function** | **Children functions** |
| {m:f.eContainer(capellacore::NamedElement).name} | {m:f.name} | {m:if f.subFunctions->size() > 0}  {m:for children | f.subFunctions}   * {m:children.name}   {m:endfor}  {m:else}  *No children function*  {m:endif} |

List of inputs:

{m:if f.incoming ->size() > 0}

| **Incoming functional exchange** | **Exchanged items** |
| --- | --- |

{m:for fe | f.incoming->filter(fa::FunctionalExchange)}

|  |  |
| --- | --- |
| {m:fe.name} | {m:if fe.exchangedItems->size() > 0}  {m:for ei | fe.exchangedItems}   * {m:ei.name}   {m:endfor}  {m:else}  *No exchanged item*  {m:endif} |

{m:endfor}

{m:else}

*No input*

{m:endif}

List of outputs:

{m:if f.outgoing->size() > 0}

| **Outgoing functional exchange** | **Exchanged items** |
| --- | --- |

{m:for fe | f.outgoing->filter(fa::FunctionalExchange)}

|  |  |
| --- | --- |
| {m:fe.name} | {m:if fe.exchangedItems->size() > 0}  {m:for ei | fe.exchangedItems}   * {m:ei.name}   {m:endfor}  {m:else}  *No exchanged item*  {m:endif} |

{m:endfor}

{m:else}

*No output*

{m:endif}

Participation to functional chains:

{m:if f.involvingFunctionalChains->size() > 0}

{m:for FC | f.involvingFunctionalChains}

* {m:FC.name}

{m:endfor}

{m:else}

*None*

{m:endif}

Allocated to:

{m:if f. allocationBlocks->size() > 0}

{m:for sys | f. allocationBlocks}

* {m:sys.name}

{m:endfor}

{m:else}

*Function not allocated*

{m:endif}

{m:endfor}

# Descriptions of Interfaces and Data

{m:for im | self.containedSystemAnalysis.ownedDataPkg.asImageByRepresentationDescriptionName('Class Diagram Blank')}

{m:im.setWidth(400)}

{m:endfor}

{m:for package | self.containedSystemAnalysis.ownedDataPkg}

## {m:package.name}

{m:if package.ownedExchangeItems->size() > 0}

Exchange items definition:

{m:for EI | package.ownedExchangeItems}

{m:EI.name.asBookmark(EI.id)}

{m:if EI.ownedElements->size() > 0}

|  |  |
| --- | --- |
| **Owned element** | **Element type** |

{m:for element |EI.ownedElements}

|  |  |
| --- | --- |
| {m:element.name} | [{m:element.type.name}] |

{m:endfor}

{m:else}

*No contained elements*

{m:endif}

{m:endfor}

{m:endif}

{m:if package.ownedCollections->size() > 0}

Collections definition:

|  |  |
| --- | --- |
| **Collection** | **Collection type** |

{m:for collection | package.ownedCollections}

|  |  |
| --- | --- |
| {m:collection.name} | [{m:collection.type.name}] |

{m:endfor}

{m:endif}

{m:if package.ownedClasses->size() > 0}

Classes definition:

{m:for class | package.ownedClasses}

{m:class.name}

{m:if class.containedProperties->size() > 0}

|  |  |
| --- | --- |
| **Contained property** | **Property type** |

{m:for prop |class.containedProperties}

|  |  |
| --- | --- |
| {m:prop.name} | [{m:prop.type.name}] |

{m:endfor}

{m:else}

*No contained property*

{m:endif}

{m:endfor}

{m:endif}

{m:if package.ownedDataTypes->size() > 0}

Data types definition:

{m:for data | package.ownedDataTypes}

* {m:data.name}

{m:endfor}

{m:endif}

{m:endfor}

{m:for package | self.containedSystemAnalysis.ownedDataPkg.eAllContents(information::DataPkg)}

## {m:package.name}

{m:if package.ownedExchangeItems->size() > 0}

Exchange items definition:

{m:for EI | package.ownedExchangeItems}

{m:EI.name.asBookmark(EI.id)}

{m:if EI.ownedElements->size() > 0}

|  |  |
| --- | --- |
| **Owned element** | **Element type** |

{m:for element |EI.ownedElements}

|  |  |
| --- | --- |
| {m:element.name} | [{m:element.type.name}] |

{m:endfor}

{m:else}

*No contained elements*

{m:endif}

{m:endfor}

{m:endif}

{m:if package.ownedCollections->size() > 0}

Collections definition:

|  |  |
| --- | --- |
| **Collection** | **Collection type** |

{m:for collection | package.ownedCollections}

|  |  |
| --- | --- |
| {m:collection.name} | [{m:collection.type.name}] |

{m:endfor}

{m:endif}

{m:if package.ownedClasses->size() > 0}

Classes definition:

{m:for class | package.ownedClasses}

{m:class.name}

{m:if class.containedProperties->size() > 0}

|  |  |
| --- | --- |
| **Contained property** | **Property type** |

{m:for prop |class.containedProperties}

|  |  |
| --- | --- |
| {m:prop.name} | [{m:prop.type.name}] |

{m:endfor}

{m:else}

*No contained property*

{m:endif}

{m:endfor}

{m:endif}

{m:if package.ownedDataTypes->size() > 0}

Data types definition:

{m:for data | package.ownedDataTypes}

* {m:data.name}

{m:endfor}

{m:endif}

{m:endfor}