последовательност прецедентов классов активности активности BETA компонент состояний объ

В начало
Что нового ?
Быстрый старт
Online Server
Запуск
F.A.Q.
Скачать
Форум
Theme
Препроцессинг
Стандартная библиотека
Hitchhiker's Guide
PDF Guide

PlantUML > Language specification > Archimate Diagram

Archimate Diagram

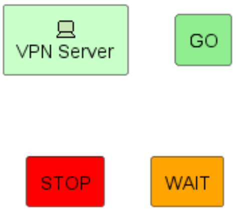
This is only a proposal and subject to change.

You are very welcome to create a new discussion on this future syntax. Your feedbacks, ideas and suggestions help us to find the right solution.

Archimate keyword

You can use the `archimate` keyword to define an element. Stereotype can optionally specify an additional icon. Some colors (`Business` , `Application` , `Motivation` , `Strategy` , `Technology` , `Physical` , `Implementation`) are also available.

```
@startuml
archimate #Technology "VPN Server" as vpnServerA ·

rectangle GO #lightgreen
rectangle STOP #red
rectangle WAIT #orange
@enduml
```



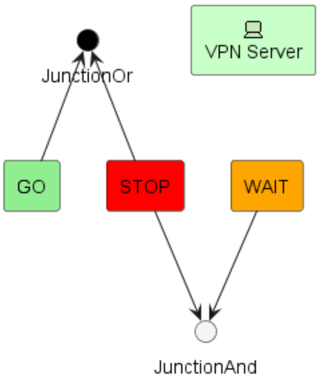
Defining Junctions

Using the `circle` keyword and the preprocessor, you can also create junctions.

последовательност прецедентов классов активности активности BETA компонент состояний объ

В начало
Что нового ?
Быстрый старт
Online Server
Запуск
F.A.Q.
Скачать
Форум
Theme
Препроцессинг
Стандартная библиотека
Hitchhiker's Guide
PDF Guide

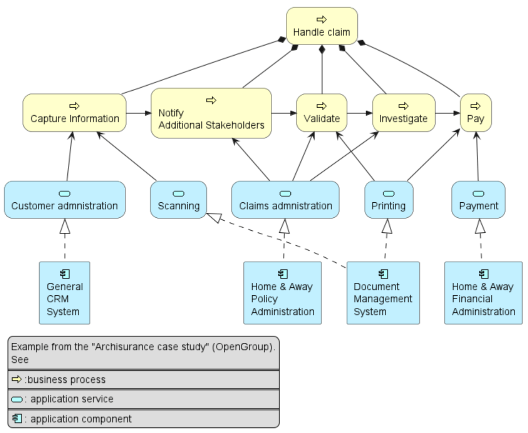
```
!define Junction_Or circle #black
!define Junction_And circle #whitesmoke

Junction_And JunctionAnd
Junction_Or JunctionOr

archimate #Technology "VPN Server" as vpnServerA ·

rectangle GO #lightgreen
rectangle STOP #red
rectangle WAIT #orange
GO -up-> JunctionOr
STOP -up-> JunctionOr
STOP -down-> JunctionAnd
WAIT -down-> JunctionAnd
@enduml
```



Example 1

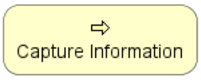
- В начало
- Что нового ?
- Быстрый старт
- Online Server
- Запуск
- F.A.Q.
- Скачать
- Форум
- Theme
- Препроцессинг
- Стандартная библиотека
- Hitchhiker's Guide
- PDF Guide

```
skinparam rectangle<<behavior>> {
        roundCorner 25
}
sprite $bProcess jar:archimate/business-process
sprite $aService jar:archimate/application-servic
sprite $aComponent jar:archimate/application-compo

rectangle "Handle claim"  as HC <<$bProcess>><<bel
rectangle "Capture Information"  as CI <<$bProces
rectangle "Notify\nAdditional Stakeholders" as NA
rectangle "Validate" as V <<$bProcess>><<behavior
rectangle "Investigate" as I <<$bProcess>><<behav
rectangle "Pay" as P <<$bProcess>><<behavior>> #Bu

HC *-down- CI
HC *-down- NAS
HC *-down- V
HC *-down- I
HC *-down- P

CI -right->> NAS
NAS -right->> V
V -right->> I
I -right->> P

rectangle "Scanning" as scanning <<$aService>><<be
rectangle "Customer admnistration" as customerAdm
rectangle "Claims admnistration" as claimsAdminis
rectangle Printing <<$aService>><<behavior>> #App
rectangle Payment <<$aService>><<behavior>> #Appl

scanning -up-> CI
customerAdministration  -up-> CI
claimsAdministration -up-> NAS
claimsAdministration -up-> V
claimsAdministration -up-> I
Payment -up-> P

Printing -up-> V
Printing -up-> P

rectangle "Document\nManagement\nSystem" as DMS <
rectangle "General\nCRM\nSystem" as CRM <<$aCompo
rectangle "Home & Away\nPolicy\nAdministration" a
rectangle "Home & Away\nFinancial\nAdministration

DMS .up.|> scanning
DMS .up.|> Printing
CRM .up.|> customerAdministration
HAPA .up.|> claimsAdministration
HFPA .up.|> Payment

legend left
Example from the "Archisurance case study" (OpenG
See
====
<$bProcess> :business process
====
<$aService> : application service
====
<$aComponent> : application component
endlegend
@enduml
```



Example 2

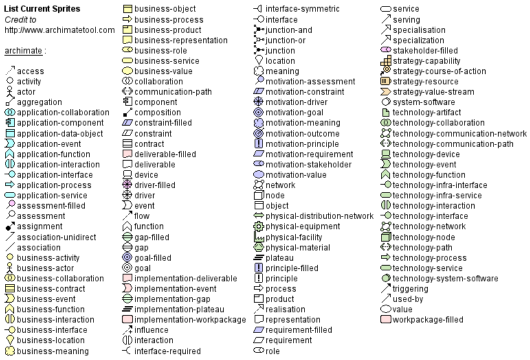
```
skinparam roundcorner 25
rectangle "Capture Information"  as CI <<$archima
@enduml
```



List possible sprites

You can list all possible sprites for Archimate using the following diagram:

```
@startuml
listsprite
@enduml
```



ArchiMate Macros

Archimate Macros and Library

A list of Archimate macros are defined Archimate-PlantUML here which simplifies the creation of ArchiMate diagrams, and Archimate is natively on the Standard Library of PlantUML.

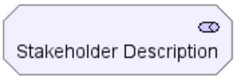
Archimate elements

Using the macros, creation of ArchiMate elements are done using the following format: `Category_ElementName(nameOfTheElement, "description")`

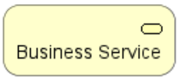
For example:

- To define a *Stakeholder* element, which is part of Motivation category, the syntax will be `Motivation_Stakeholder(StakeholderElement, "Stakeholder Description")`:

```
@startuml
!include <archimate/Archimate>
Motivation_Stakeholder(StakeholderElement, "Stakel
@enduml
```



```
@startuml
!include <archimate/Archimate>
Business_Service(BService, "Business Service")
@enduml
```



Archimate relationships

The ArchiMate relationships are defined with the following pattern: `Rel_RelationType(fromElement, toElement, "description")` and to define the direction/orientation of the two elements: `Rel_RelationType_Direction(fromElement, toElement, "description")`

The `RelationTypes` supported are:

- Access
- Aggregation
- Assignment
- Association
- Composition
- Flow
- Influence
- Realization
- Serving
- Specialization
- Triggering

The `Directions` supported are:

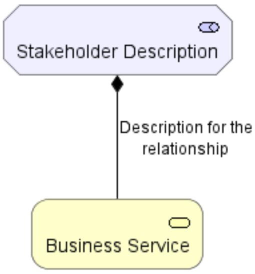
- Up
- Down
- Left
- Right

For example:

- To denote a composition relationship between the *Stakeholder* and *Business Service* defined above, the syntax will be

`Rel_Composition(StakeholderElement, BService, "Description for the relationship")`

```
@startuml
!include <archimate/Archimate>
Motivation_Stakeholder(StakeholderElement, "Stakel
Business_Service(BService, "Business Service")
Rel_Composition(StakeholderElement, BService, "De:
@enduml
```



- Unordered List ItemTo orient the two elements in top - down position, the syntax will be

`Rel_Composition_Down(StakeholderElement, BService, "Description for the relationship")`

последовательност прецедентов классов активности активности BETA компонент состояний объ

В начало

Что нового ?

Быстрый старт

Online Server

Запуск

F.A.Q.

Скачать

Форум

Theme

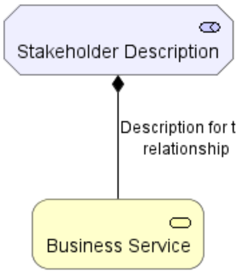
Препроцессинг

Стандартная библиотека

Hitchhiker's Guide

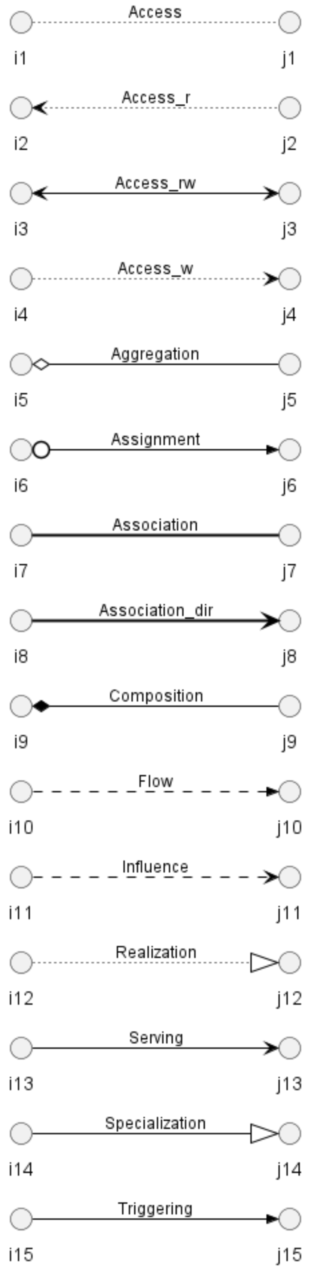
PDF Guide

```
!include <archimate/Archimate>
Motivation_Stakeholder(StakeholderElement, "Stakel
Business_Service(BService, "Business Service")
Rel_Composition_Down(StakeholderElement, BService
@enduml
```



Appendice: Examples of all Archimate RelationTypes

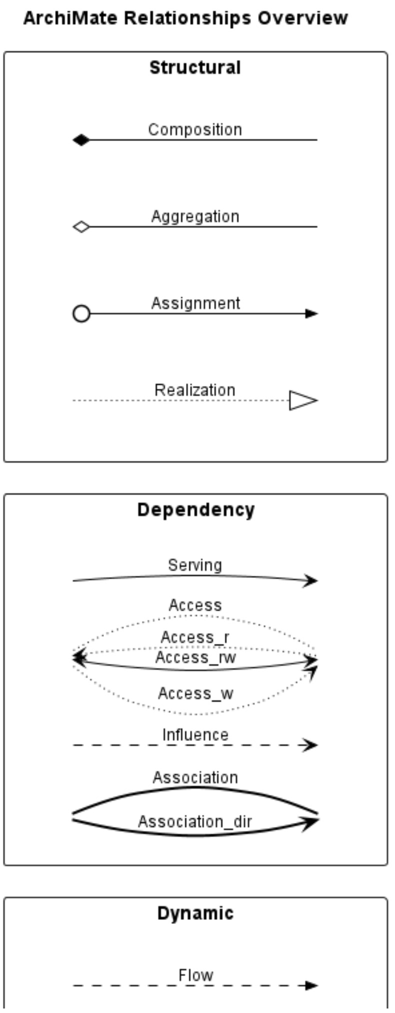
```
left to right direction
skinparam nodesep 4
!include <archimate/Archimate>
Rel_Triggering(i15, j15, Triggering)
Rel_Specialization(i14, j14, Specialization)
Rel_Serving(i13, j13, Serving)
Rel_Realization(i12, j12, Realization)
Rel_Influence(i11, j11, Influence)
Rel_Flow(i10, j10, Flow)
Rel_Composition(i9, j9, Composition)
Rel_Association_dir(i8, j8, Association_dir)
Rel_Association(i7, j7, Association)
Rel_Assignment(i6, j6, Assignment)
Rel_Aggregation(i5, j5, Aggregation)
Rel_Access_w(i4, j4, Access_w)
Rel_Access_rw(i3, j3, Access_rw)
Rel_Access_r(i2, j2, Access_r)
Rel_Access(i1, j1, Access)
@enduml
```

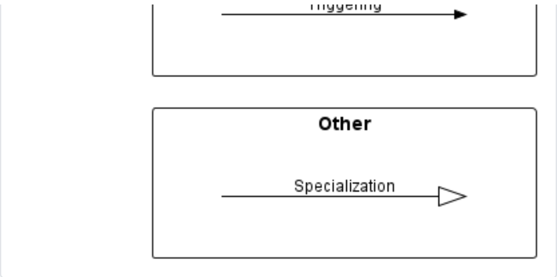


```
@startuml
title ArchiMate Relationships Overview
skinparam nodesep 5
<style>
interface {
    shadowing 0
    backgroundcolor transparent
    linecolor transparent
    FontColor transparent
}
</style>
!include <archimate/Archimate>
left to right direction
```

В начало
Что нового ?
Быстрый старт
Online Server
Запуск
F.A.Q.
Скачать
Форум
Theme
Препроцессинг
Стандартная библиотека
Hitchhiker's Guide
PDF Guide

```
() j14
}


rectangle Dynamic {
() i10
() j10
() i15
() j15
}

rectangle Dependency {
() i13
() j13
() i4
() j4
() i11
() j11
() i7
() j7
}

rectangle Structural {
() i9
() j9
() i5
() j5
() i6
() j6
() i12
() j12
}

Rel_Triggering(i15, j15, Triggering)
Rel_Specialization(i14, j14, Specialization)
Rel_Serving(i13, j13, Serving)
Rel_Realization(i12, j12, Realization)
Rel_Influence(i11, j11, Influence)
Rel_Flow(i10, j10, Flow)
Rel_Composition(i9, j9, Composition)
Rel_Association_dir(i7, j7, \nAssociation_dir)
Rel_Association(i7, j7, Association)
Rel_Assignment(i6, j6, Assignment)
Rel_Aggregation(i5, j5, Aggregation)
Rel_Access_w(i4, j4, Access_w)
Rel_Access_rw(i4, j4, Access_rw)
Rel_Access_r(i4, j4, Access_r)
Rel_Access(i4, j4, Access)
@enduml
```





[Adapted from Archimate PR#25]