



Exploring - Relationships

This document explores all aspects of how eaDocX can help you see the relationships within your EA model.

Section: Section1

1 Basic Relationships

This section shows how different formatting options can be chosen to show more or less of the relationships between elements.

In all cases, you need to tell eaDocX how the two elements are related:

- What type of connector links them (or 'Any')
 - What type of element is at the other end of the connector (or 'Any') - this is the 'Target element'
 - The direction of the relationship, or 'Either' if you don't know or don't care.
 - and optionally the name and/or stereotype of the connector, and the stereotype of the target element
1. **Just a single attribute of a related element.** eaDocX calls the 'Relationship Elements', and they can be just one attribute or a related element, or of the connector which links them. If there are many elements or connectors which fit the specification, they are returned in a comma- or line-separated list.

These can be printed as a column in a table, or as an Inline paragraph

- **A Table of related elements.** This finds all the elements which fit the criteria you specify, then prints them in a table. Because this creates a table, it's only available where the source element is printed Inline: eaDocX won't print tables inside other tables. eaDocX calls these **Relationship Tables**.
- **A Relationship Element.** This is similar to (2), except that the target elements will print according to their Profile.

To illustrate this, we have chosen some different stereotypes of Feature, related to some stereotypes of 'Requirement'.

1.1 Relationship Attributes

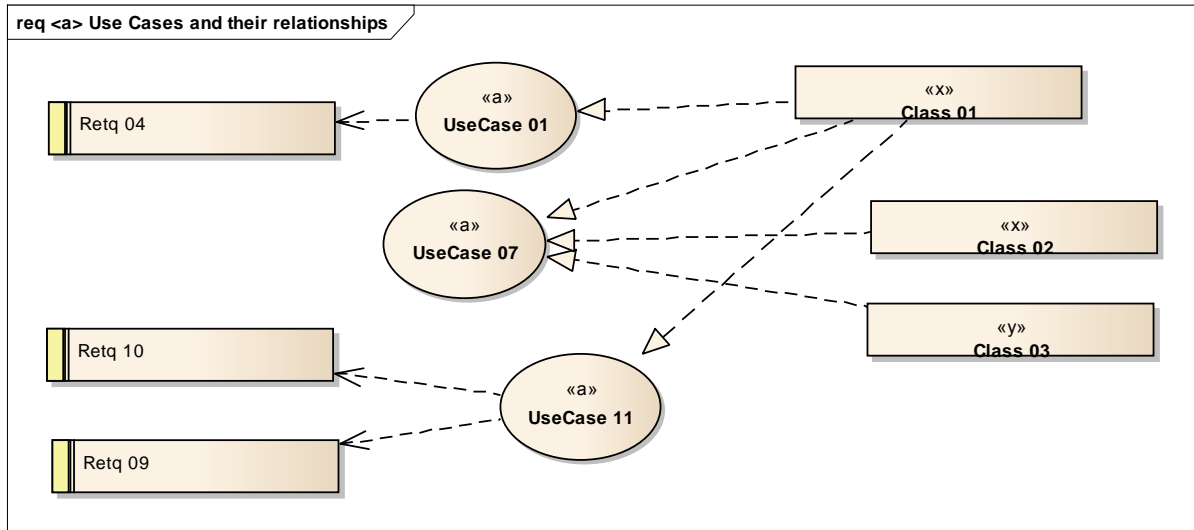


Figure 1 : <a> Use Cases and their relationships

This section prints the <<a>>Use Cases INLINE, then some of the related Class and Requirement elements.

1.1.1 UseCase 01

Alias: UC01

Stereotype: a

This data comes from Relationship Attributes

Dependent Requirements: [Retq 04](#)

Realised by <>Classes: [Class 01](#)

1.1.2 UseCase 07

Alias: UC07

Stereotype: a

This data comes from Relationship Attributes

Dependent Requirements: none

Realised by <>Classes: [Class 02](#), [Class 01](#)

1.1.3 UseCase 11

Alias: UC11

Stereotype: a

This data comes from Relationship Attributes

Dependent Requirements: [Retq 10](#), [Retq 09](#)

Realised by <>Classes: [Class 01](#)

1.2 Relationship tables

For these <>Use Cases, we will include more information about their related elements.

Each one will have a table of their related Classes.

This means that there will be some duplication of data, but that's OK because all the data comes from the model, so we are not creating ourselves a maintenance problem.

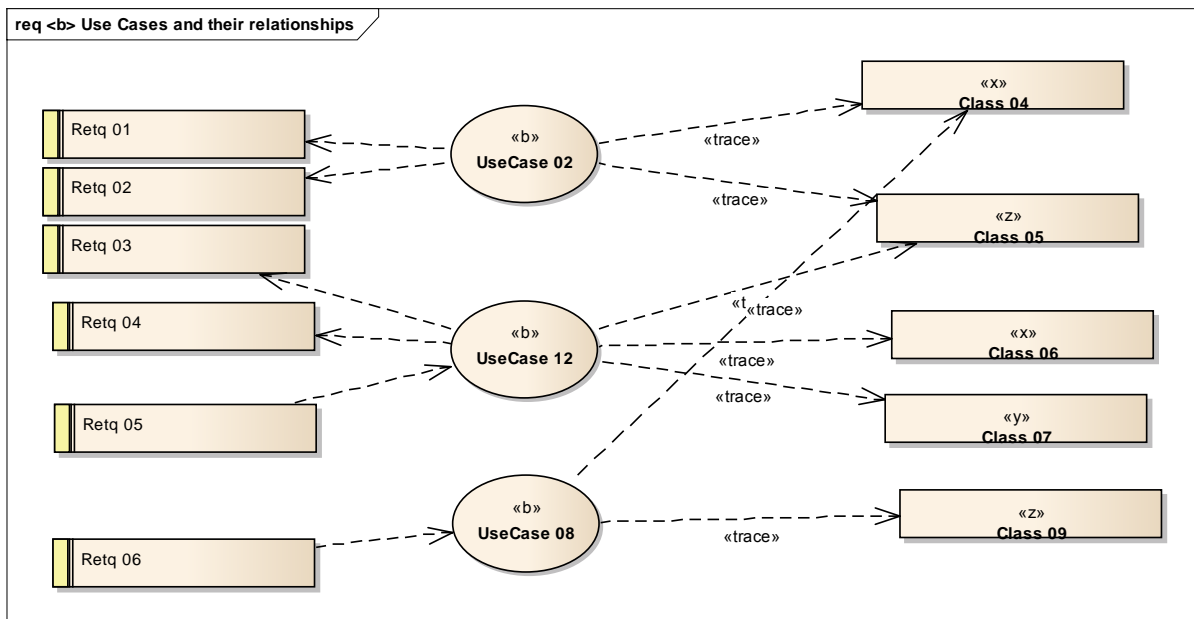


Figure 2 : Use Cases and their relationships

Note that in this diagram, the Dependency relationships between Use Cases and Requirements are in two different directions.

1.2.1 UseCase 02

Alias: UC02

1.2.1.1 Requirements related by a forward-dependency

Name	Stereotype	Element Type
Retq 01	red	Requirement
Retq 02	blue	Requirement

1.2.1.2 All related requirements

Name	Stereotype	Element Type
Retq 01	red	Requirement
Retq 02	blue	Requirement

1.2.2 UseCase 08

Alias: UC08

1.2.2.1 Requirements related by a backwards dependency

Name	Stereotype	Element Type
Retq 06	green	Requirement

1.2.2.2 All related requirements

Name	Stereotype	Element Type
Retq 06	green	Requirement

1.2.3 UseCase 12

Alias: UC12

1.2.3.1 Requirements related by a forward-dependency

Name	Stereotype	Element Type
Retq 03	green	Requirement
Retq 04	red	Requirement

1.2.3.2 Requirements related by a backwards dependency

Name	Stereotype	Element Type
Retq 05	blue	Requirement

1.2.3.3 All related requirements

Name	Stereotype	Element Type
Retq 03	green	Requirement
Retq 04	red	Requirement
Retq 05	blue	Requirement

1.3 Relationship Tables (2)

In this example, we show how you can print attributes of the connector, as well as tagged values of the connector and the target element.

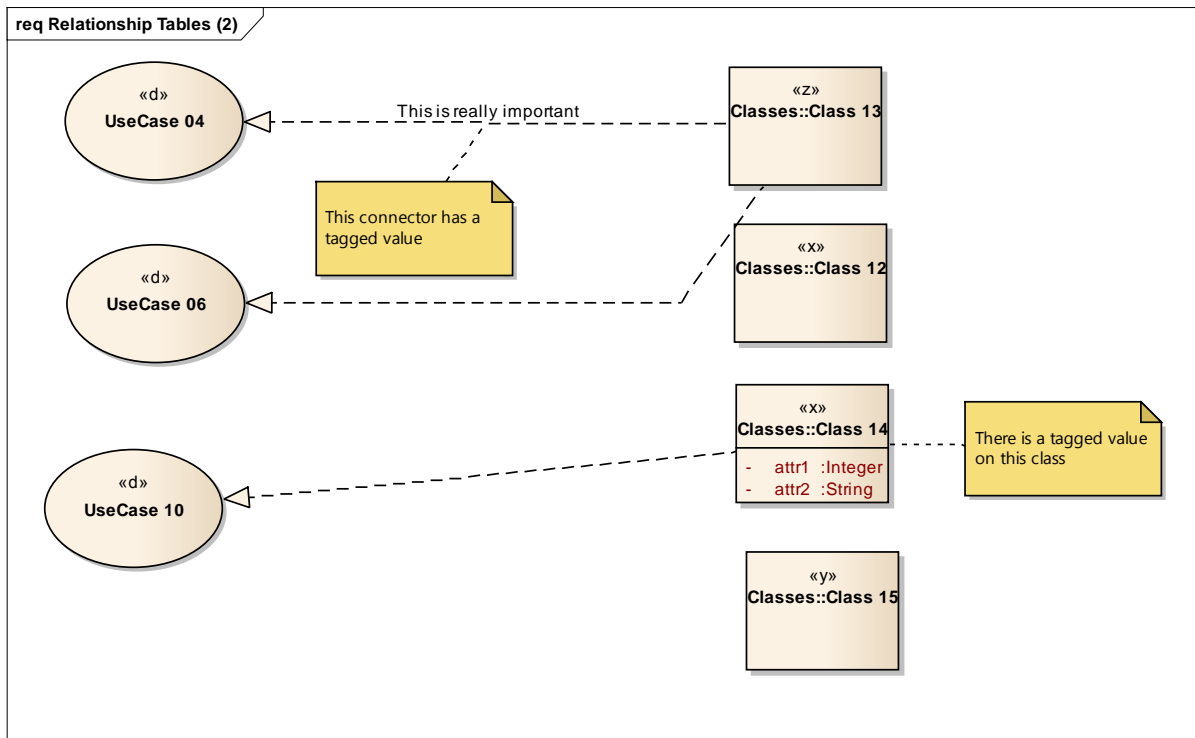


Figure 3 : Relationship Tables (2)

1.3.1 UseCase 04

Alias: UC04

1.3.1.1 Related classes

Name - hyperlinked	classTV tagged value of relationship	Relationship Type	relTV tagged value of class
Class 13		Realisation	This is a tagged value of the connector

1.3.2 UseCase 06

Alias: UC06

1.3.2.1 Related classes

Name - hyperlinked	classTV tagged value of relationship	Relationship Type	relTV tagged value of class
Class 13		Realisation	

1.3.3 UseCase 10

Alias: UC10

1.3.3.1 Related classes

Name - hyperlinked	classTV tagged value of relationship	Relationship Type	relTV tagged value of class
Class 14	This is a tagged value of a class	Realisation	This is another tagged value of a connector

1.4 Relationship Elements

Relationship elements allow you to print your documents without using the Package structure at all. In this example, we have connected just one <<c>>Use Case to some other elements, which already have Profiles defined for this document.

You can see that this is very simple to specify: in this case, the <<c>>Use Case is linked to some Packages, so we rely on the Profile already defined elsewhere to print the Package, and anything which we find in it.

An alternative way to get away from the Package structure is to print the content of a diagram, or use an Element Report.

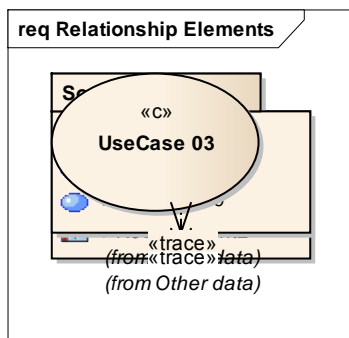


Figure 4 : Relationship Elements

1.4.1 UseCase 03

Alias: UC03

1.4.1.1 Related Stuff

1.4.1.1.1 Some stuff

ElementType	Stereotype	Name	Alias
Requirement	red	Retq 10	REQ10

1.4.1.1.1.1 UseCase 05

Alias: UC05

1.4.1.1.1.1 Related Stuff

1.4.1.1.1.2 UseCase 09

Alias: UC09

1.4.1.1.1.2.1 Related Stuff

1.4.1.1.2 TinyMatrix

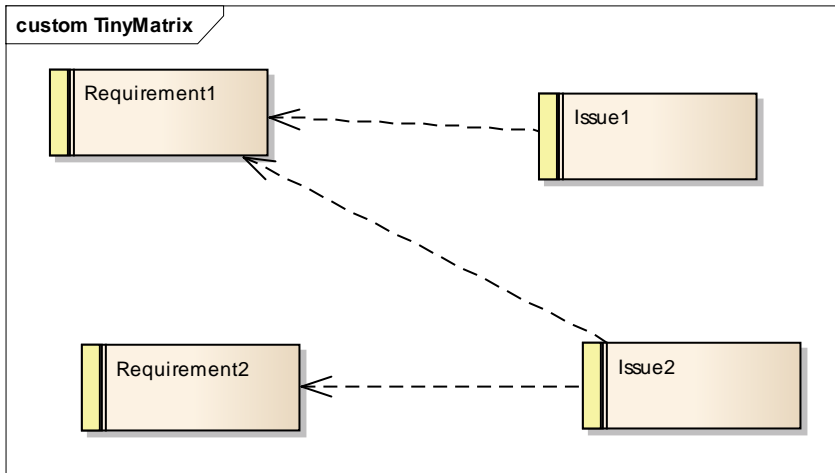


Figure 5 : TinyMatrix

ElementType	Stereotype	Name	Alias
Requirement		Requirement1	
Requirement		Requirement2	

1.5 Other data

1.5.1 Some stuff

ElementType	Stereotype	Name	Alias
Requirement	red	Retq 10	REQ10

1.5.1.1 UseCase 05

Alias: UC05

1.5.1.1.1 Related Stuff

1.5.1.2 UseCase 09

Alias: UC09

1.5.1.2.1 Related Stuff

1.5.2 Classes

Element Type	Name	ParentStereotype
Class	Class 01	
Class	Class 02	
Class	Class 03	
Class	Class 04	
Class	Class 05	
Class	Class 06	
Class	Class 07	
Class	Class 08	
Class	Class 09	
Class	Class 10	
Class	Class 11	
Class	Class 12	
Class	Class 13	
Class	Class 14	
Class	Class 15	

1.5.3 Requirements

ElementType	Stereotype	Name	Alias
Requirement	red	Retq 01	REQ01
Requirement	blue	Retq 02	REQ02
Requirement	green	Retq 03	REQ03
Requirement	red	Retq 04	REQ04
Requirement	blue	Retq 05	REQ05
Requirement	green	Retq 06	REQ06
Requirement	red	Retq 07	REQ07

ElementType	Stereotype	Name	Alias
Requirement	blue	Retq 08	REQ08
Requirement	green	Retq 09	REQ09

End

[Section: Introduction](#)

2 Relationships

As your EA models get more complex, and your network of elements gets more heavily linked, you may want to show elements which are more than one 'hop' away. ('Hop' here is just shorthand for a single connector')

Displaying multi-hop links can be a really powerful way to show information, but it does mean that your model needs to keep to a strict meta-model.

2.1 Basic Relationships

This section shows how different formatting options can be chosen to show more or less of the relationships between elements.

In all cases, you need to tell eaDocX how the two elements are related:

- What type of connector links them (or 'Any')
 - What type of element is at the other end of the connector (or 'Any') - this is the 'Target element'
 - The direction of the relationship, or 'Either' if you don't know or don't care.
 - and optionally the name and/or stereotype of the connector, and the stereotype of the target element
1. **Just a single attribute of a related element.** eaDocX calls the 'Relationship Elements', and they can be just one attribute or a related element, or of the connector which links them. If there are many elements or connectors which fit the specification, they are returned in a comma- or line-separated list.

These can be printed as a column in a table, or as an Inline paragraph

- **A Table of related elements.** This finds all the elements which fit the criteria you specify, then prints them in a table. Because this creates a table, it's only available where the source element is printed Inline: eaDocX won't print tables inside other tables. eaDocX calls these **Relationship Tables**.

- **A Relationship Element.** This is similar to (2), except that the target elements will print according to their Profile.

To illustrate this, we have chosen some different stereotypes of Feature, related to some stereotypes of 'Requirement'.

2.1.1 Relationship Attributes

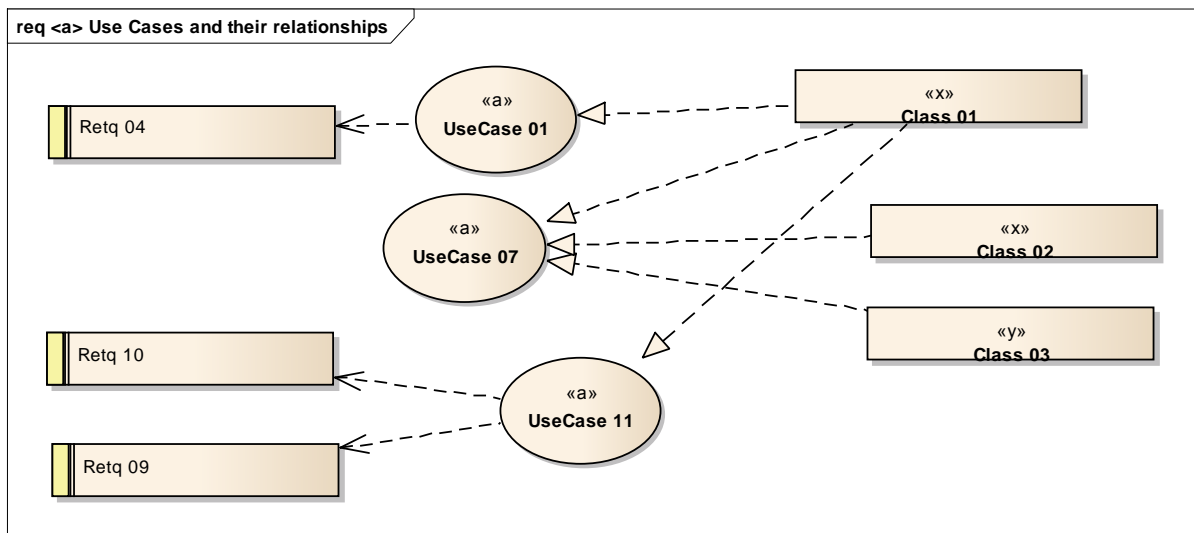


Figure 6 : <a> Use Cases and their relationships

This section prints the <<a>>Use Cases INLINE, then some of the related Class and Requirement elements.

2.1.1.1 UseCase 01

Alias: UC01

Stereotype: a

This data comes from Relationship Attributes

Dependent Requirements: [Retq 04](#)

Realised by <>Classes: [Class 01](#)

2.1.1.2 UseCase 07

Alias: UC07

Stereotype: a

This data comes from Relationship Attributes

Dependent Requirements: none

Realised by <>Classes: [Class 02](#), [Class 01](#)

2.1.1.3 UseCase 11

Alias: UC11

Stereotype: a

This data comes from Relationship Attributes

Dependent Requirements: [Retq 10](#), [Retq 09](#)

Realised by <>Classes: [Class 01](#)

2.1.2 Relationship tables

For these <>Use Cases, we will include more information about their related elements.

Each one will have a table of their related Classes.

This means that there will be some duplication of data, but that's OK because all the data comes from the model, so we are not creating ourselves a maintenance problem.

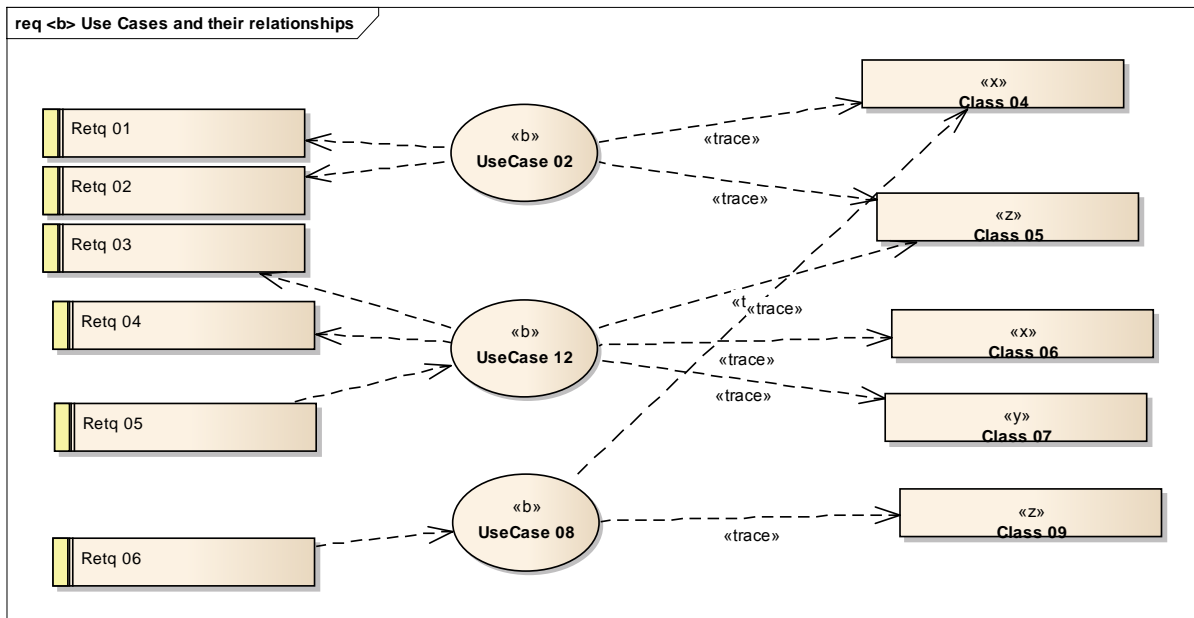


Figure 7 : Use Cases and their relationships

Note that in this diagram, the Dependency relationships between Use Cases and Requirements are in two different directions.

2.1.2.1 UseCase 02

Alias: UC02

2.1.2.1.1 Requirements related by a forward-dependency

Name	Stereotype	Element Type
Retq 01	red	Requirement

Name	Stereotype	Element Type
Retq 02	blue	Requirement

2.1.2.1.2 All related requirements

Name	Stereotype	Element Type
Retq 01	red	Requirement
Retq 02	blue	Requirement

2.1.2.2 UseCase 08

Alias: UC08

2.1.2.2.1 Requirements related by a backwards dependency

Name	Stereotype	Element Type
Retq 06	green	Requirement

2.1.2.2.2 All related requirements

Name	Stereotype	Element Type
Retq 06	green	Requirement

2.1.2.3 UseCase 12

Alias: UC12

2.1.2.3.1 Requirements related by a forward-dependency

Name	Stereotype	Element Type
Retq 03	green	Requirement
Retq 04	red	Requirement

2.1.2.3.2 Requirements related by a backwards dependency

Name	Stereotype	Element Type
Retq 05	blue	Requirement

2.1.2.3.3 All related requirements

Name	Stereotype	Element Type
Retq 03	green	Requirement
Retq 04	red	Requirement
Retq 05	blue	Requirement

2.1.3 Relationship Tables (2)

In this example, we show how you can print attributes of the connector, as well as tagged values of the connector and the target element.

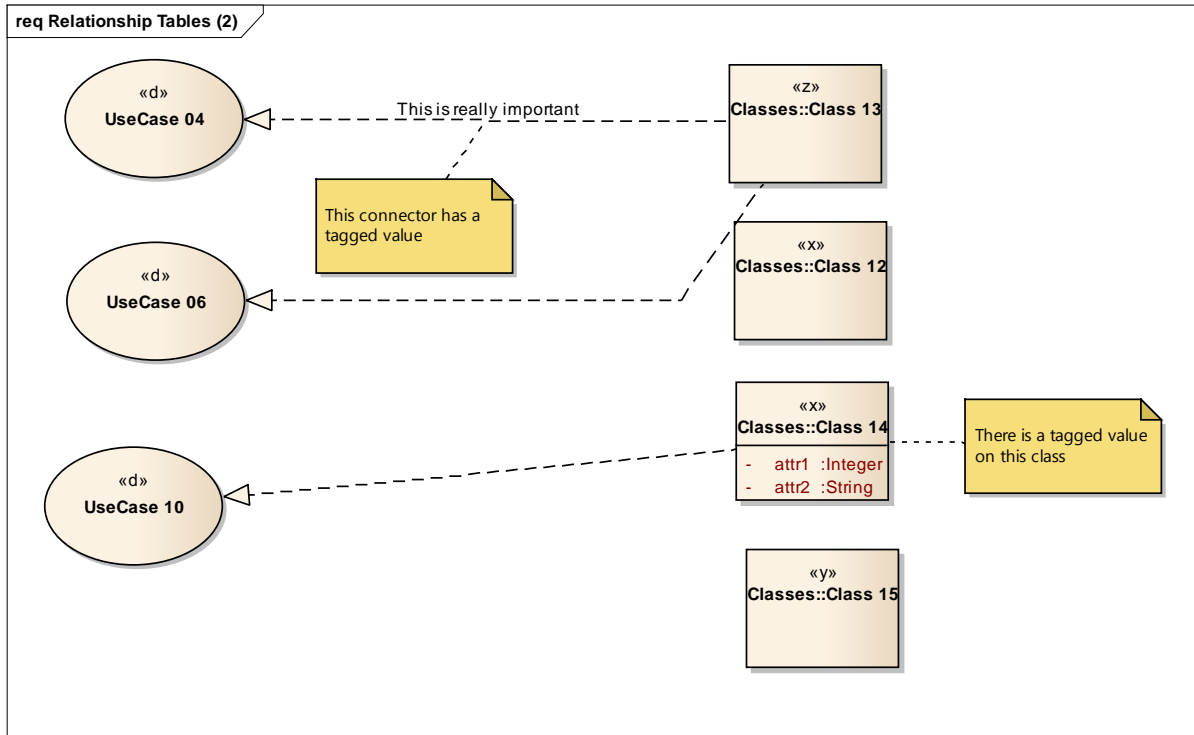


Figure 8 : Relationship Tables (2)

2.1.3.1 UseCase 04

Alias: UC04

2.1.3.1.1 Related classes

Name - hyperlinked	classTV tagged value of relationship	Relationship Type	relTV tagged value of class
Class 13		Realisation	This is a tagged value of the connector

2.1.3.2 UseCase 06

Alias: UC06

2.1.3.2.1 Related classes

Name - hyperlinked	classTV tagged value of relationship	Relationship Type	relTV tagged value of class
Class 13		Realisation	

2.1.3.3 UseCase 10

Alias: UC10

2.1.3.3.1 Related classes

Name - hyperlinked	classTV tagged value of relationship	Relationship Type	relTV tagged value of class
Class 14	This is a tagged value of a class	Realisation	This is another tagged value of a connector

2.1.4 Relationship Elements

Relationship elements allow you to print your documents without using the Package structure at all. In this example, we have connected just one <<c>>Use Case to some other elements, which already have Profiles defined for this document.

You can see that this is very simple to specify: in this case, the <<c>>Use Case is linked to some Packages, so we rely on the Profile already defined elsewhere to print the Package, and anything which we find in it.

An alternative way to get away from the Package structure is to print the content of a diagram, or use an Element Report.

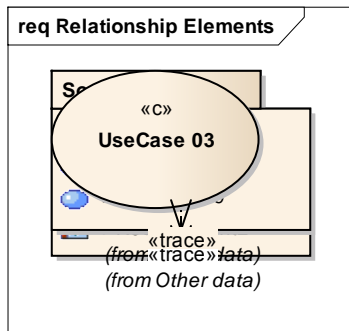


Figure 9 : Relationship Elements

2.1.4.1 UseCase 03

Alias: UC03

2.1.4.1.1 Related Stuff

2.1.4.1.1.1 Some stuff

ElementType	Stereotype	Name	Alias
Requirement	red	Retq 10	REQ10

2.1.4.1.1.1.1 UseCase 05

Alias: UC05

2.1.4.1.1.1.1.1 Related Stuff

2.1.4.1.1.1.2 UseCase 09

Alias: UC09

2.1.4.1.1.1.2.1 Related Stuff

2.1.4.1.1.2 TinyMatrix

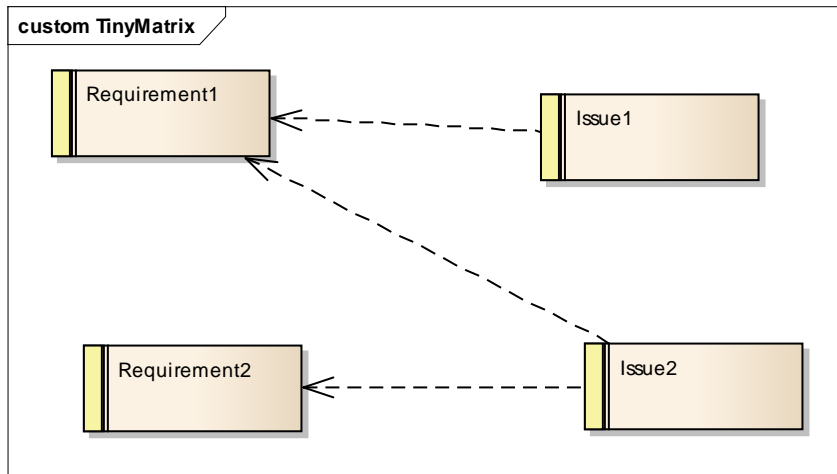


Figure 10 : TinyMatrix

ElementType	Stereotype	Name	Alias
Requirement		Requirement1	
Requirement		Requirement2	

2.1.5 Other data

2.1.5.1 Some stuff

ElementType	Stereotype	Name	Alias
Requirement	red	Retq 10	REQ10

2.1.5.1.1 UseCase 05

Alias: UC05

2.1.5.1.1.1 Related Stuff

2.1.5.1.2 UseCase 09

Alias: UC09

2.1.5.1.2.1 Related Stuff

2.1.5.2 Classes

Element Type	Name	ParentStereotype
Class	Class 01	
Class	Class 02	
Class	Class 03	
Class	Class 04	
Class	Class 05	
Class	Class 06	
Class	Class 07	
Class	Class 08	
Class	Class 09	
Class	Class 10	
Class	Class 11	
Class	Class 12	
Class	Class 13	
Class	Class 14	
Class	Class 15	

2.1.5.3 Requirements

ElementType	Stereotype	Name	Alias
Requirement	red	Retq 01	REQ01
Requirement	blue	Retq 02	REQ02
Requirement	green	Retq 03	REQ03
Requirement	red	Retq 04	REQ04
Requirement	blue	Retq 05	REQ05
Requirement	green	Retq 06	REQ06
Requirement	red	Retq 07	REQ07
Requirement	blue	Retq 08	REQ08
Requirement	green	Retq 09	REQ09

2.2 Some Data

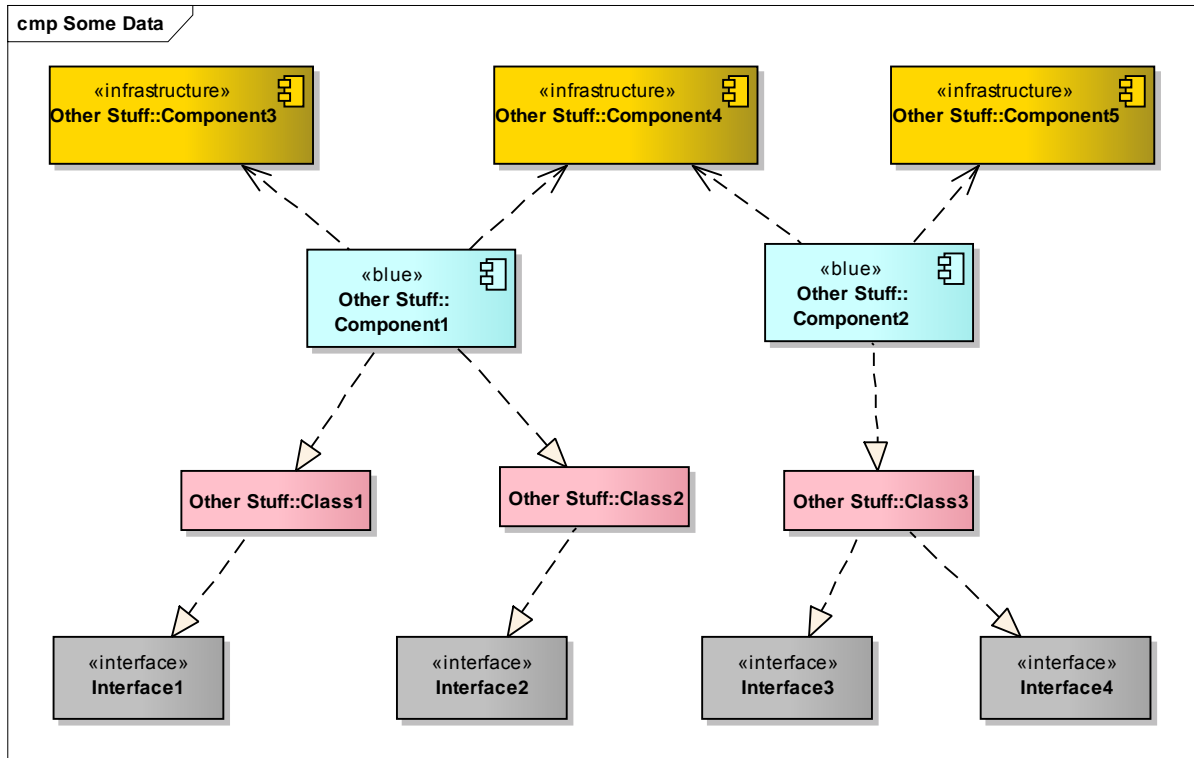


Figure 11 : Some Data

2.2.1 Interface1

Number of link hops: 1

Realised by Class: [Class1](#)

Number of link hops: 2

Realised by Blue Component: [Component1](#)

Number of link hops: 3

Needs Infrastructure Components: [Component3](#), [Component4](#)

2.2.2 Interface2

Number of link hops: 1

Realised by Class: [Class2](#)

Number of link hops: 2

Realised by Blue Component: [Component1](#)

Number of link hops: 3

Needs Infrastructure Components: [Component3](#), [Component4](#)

2.2.3 Interface3

Number of link hops: 1

Realised by Class: [Class3](#)

Number of link hops: 2

Realised by Blue Component: [Component2](#)

Number of link hops: 3

Needs Infrastructure Components: [Component4](#), [Component5](#)

2.2.4 Interface4

Number of link hops: 1

Realised by Class: [Class3](#)

Number of link hops: 2

Realised by Blue Component: [Component2](#)

Number of link hops: 3

Needs Infrastructure Components: [Component4](#), [Component5](#)

End

Section: RefComponents

2.3 Referenced Components

These are the Components which are referenced in the document above.

Element Type	Component	Stereotype	Description
Component	Component1	blue	
Component	Component2	blue	
Component	Component3	infrastructure	
Component	Component4	infrastructure	
Component	Component5	infrastructure	

End

Section: RefClasses

2.4 Referenced Classes

Element Type	Name	ParentStereotype
Class	Class1	
Class	Class2	
Class	Class3	

Element Type	Name	Parent Stereotype
Class	Class 01	
Class	Class 02	
Class	Class 13	
Class	Class 14	

End

Section: Special

3 Special Relationships

Some kinds of EA relationship have special characteristics.

- All can have TaggedValues attached to them, and the example below shows how these can be printed
- InformationFlow relationships can also have additional information attached to them - the 'Information Item'
- In State Diagrams, the state transitions between states can also have additional information attached to them, such as details of the guard conditions.

3.1 Information Flows

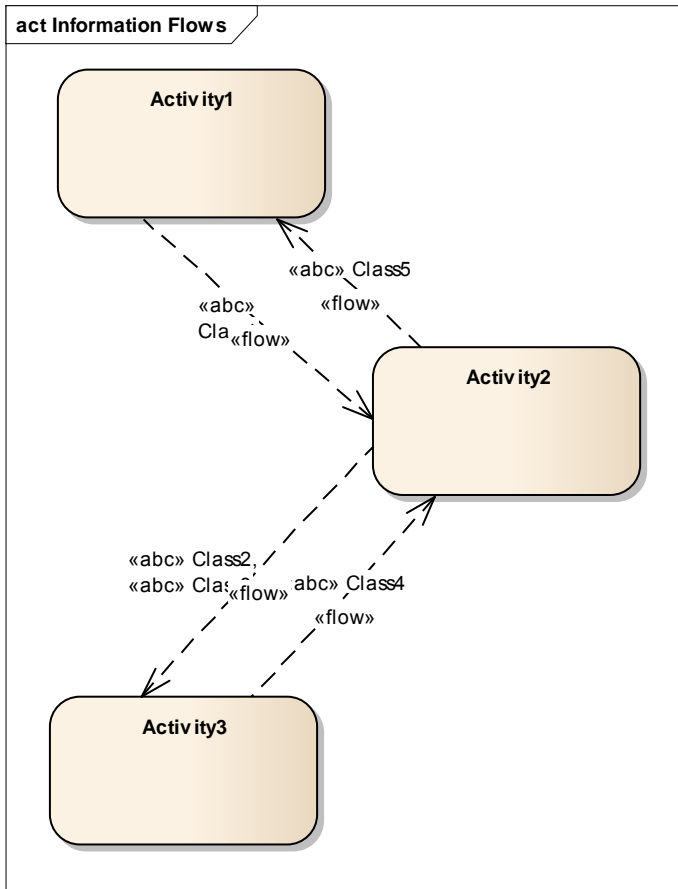


Figure 12 : Information Flows

Name	Next Activity	Information Items - FWD	Info Items - reverse
Activity1	Activity2	Class1	Class5
Activity2	Activity3 , Activity1	Class2 , Class3 , Class5	Class1 , Class4
Activity3	Activity2	Class4	Class2 , Class3

3.2 State Diagram

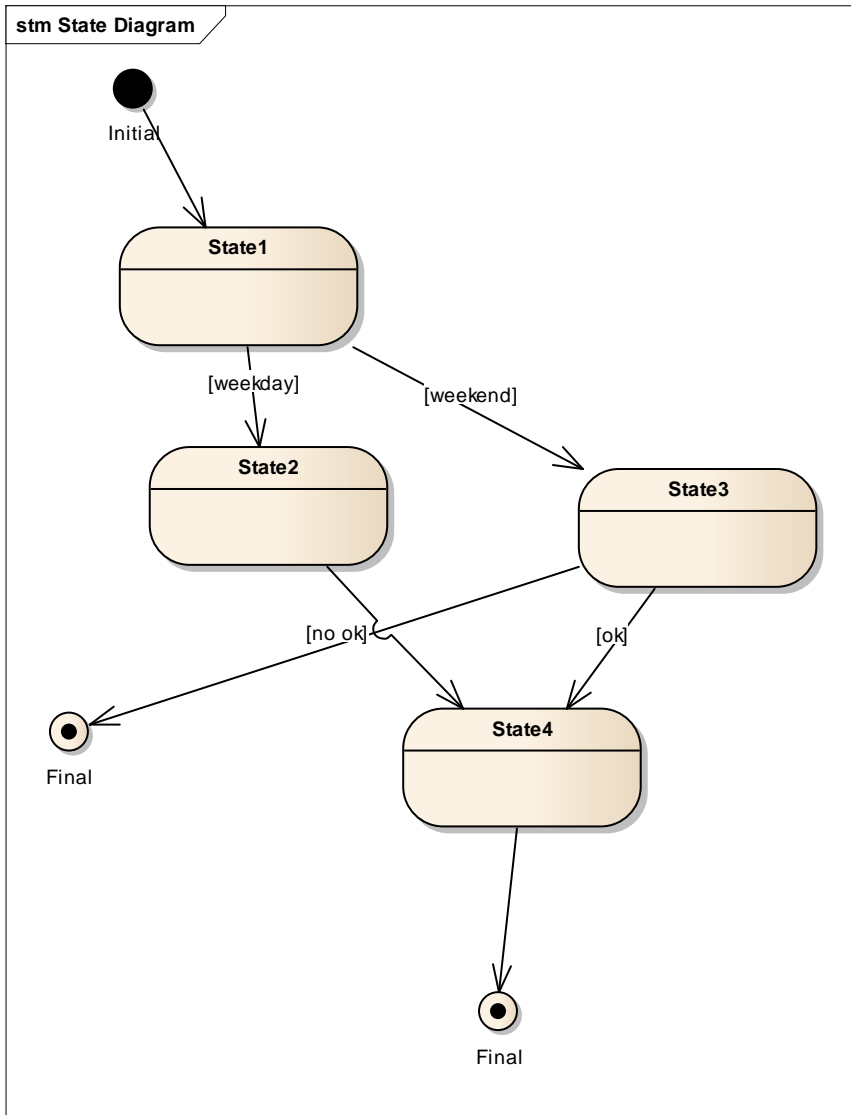


Figure 13 : State Diagram

State	Previous State	Next state	guard	[guard] Next state
State4	State2 , State3	Final	ok,	[ok] State3 ,
State1	Initial	State2 , State3	weekday, weekend	[weekday] State2 , [weekend] State3
State2	State1	State4	weekday,	[weekday] State1 ,
State3	State1	State4 , Final	weekend, ok, no ok	[weekend] State1 , [ok] State4 , [no ok] Final

3.3 Supporting Data

This package has the objects which are referenced in the Information Flow diagrams above.

Element Type	Class
Class	Class1
Class	Class2
Class	Class3
Class	Class4
Class	Class5

End

