|  |
| --- |
|  |
| Version 1.4 |

Image with outdoor object

Auto-generated description

manual

demo4 mdg

Explanation of installation and use of the Sparx Enterprise Architect MDG



## Table of contents

Table of contents 2

Download required files MDG 4

Import and configure MDG 5

Installing MDG 5

MDG Verification 6

Scripts 8

Importing scripts 8

Verifying import scripts 9

Rename TPD elements 10

Clean-up DEMO model 11

Update ARSn 12

Value Types 13

Importeren predefined tagged values 14

Operation DEMO4 MDG 15

Introduction 15

Cooperation Model (CM Cooperation Model) 15

Process Model (PM procesmodel) 22

Fact Model (FM feitenmodel) 23

Action Model 24

Implementation matrix (IMX) 34

Elaboration of Social Housing as an example 36

Introduction 36

Transaction Product Table 36

Coordination Structure Diagram 36

Auxiliary diagrams (OER rounds) 37

Process model 47

ARS 01.2 49

ARS 01.3 49

ARS 01.4 50

ARS 02.5 50

ARS 03.2 51

ARS 03.4 52

ARS 03.5 53

ARS 04.10 54

ARS 04.11 55

ARS 04.12 55

ARS 04.13 56

FM complete 57

## Download required files MDG

Go to the EAxpertise site. On the homepage, select the DEMO4 MDG: https://eaxpertise.nl/cmsform.aspx?webpage=demo4mdg.

The required downloads consist of three parts;

* MDG, an XML file that can be imported as a technology into a Sparx model
* Script import, an XML file that can be imported as reference data. It consists of three scripts that are useful when creating a DEMO4 model
* Templates, an XML file that can be imported as an import file, which creates a package with useful templates.

Afbeelding met tekst, schermopname, software, nummer

Automatisch gegenereerde beschrijving

Figure 1: list of download files

Save the files in a discoverable place. In this guide, we will refer to this path <DEMO4 downloads>.

## Import and configure MDG

### Installing MDG

Open the model where the MDG is to be imported.

Go to the Resources explore window.

Most users have these with the windows that are already available. If not, go to "Start", "Portals", "Windows".

A selection screen with Portals appears. Within the explore map you will find resources. With a right click on MDG Technology you get the option to "Import Technology".

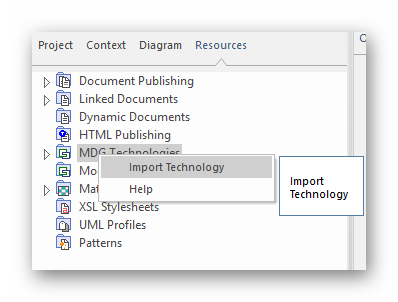


Figure 2: import MDG technology

A popup window will appear with the option to go to <DEMO4 downloads> and select file "DEMO4\_MDG.XML". Click on "open".

Image with text

Auto-generated description

Figure 3: MDG Import Window

At the time of writing this guide, the latest MDG version is version 1.3, release date 24-11-2023. The latest version will always be available on the site. Version 1.3 requires at least the Sparx EA version 16.1 to work properly.

### MDG Verification

After the MDG has been imported, we can verify whether it has succeeded and whether the MDG does what we expect it to do.

Select "Specialize", "Manage Technologies".

You will get a list of all the technologies present. By selecting DEMO4 from the list, you will be shown the details, which will allow you to see if it is the correct version with the correct release date.

Image with Text, Screen Recording, Software, Computer Icon

Auto-generated description

Figure 4: manage technologies

The effect of the MDG only becomes apparent when we create a new diagram in a suitable package. Right-click on the package in question, "add diagram".

Image with Text, Screen Recording, Software, Computer Icon

Auto-generated description

Figure 5: the available diagrams of the latest DEMO4 MDG

If DEMO4 diagrams are not immediately visible, it may be necessary to select "All Specialized" at Type. "DEMO4\_MDG\_Diagram" is the profile of the different diagrams of DEMO4. Try it out and see if the correct toolbox is shown.

## Scripts

### Importing scripts

DEMO4 Scripts.XML is a file that imports a trio of scripts that supports modeling with DEMO4. The explanation of how to use these files is covered in chapter <chapter name> of this manual.

Select "Settings", "Transfer", "Import Reference Data".

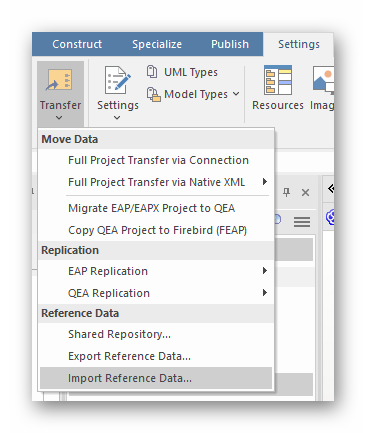


Figure 6: import scripts as reference data

In the next popup window, select the <DEMO4 downloads> location and the "DEMO4 scripts.XML" file. Then select "Automation Scripts" so that the background of this text turns blue.

Image with text

Auto-generated description

Figure 7: select "Automation Scripts"

And finally, select "Import". The scripts are imported and the message is shown that this has been done successfully with the message "Import complete".

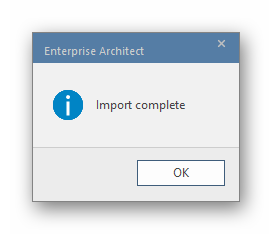


Figure 8: successful notification

### Verifying import scripts

The scripts are browser scripts. This means that the correct scripts must be selected in the browser and with a right mouse click, Specialize and Scripts the currently available scripts are displayed.

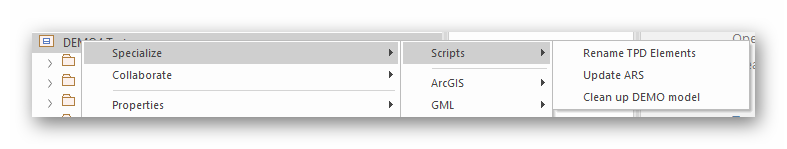


Figure 9: script "Rename TPD Elements"

### Rename TPD elements

Select a TPD composite chart under a Transaction Kind. With right-click, Specialize and Scripts you can select the "Rename TPD Elements". A system output popup window opens and asks for "Rename TKXX to TKid", where the ID must be entered.

Image with text, screenshot, number, software

Auto-generated description

Figure : rename TPD elements script

Image with text, screenshot, ride

Auto-generated description

Figure 11: In the input screen, you can enter the new ID

All TPD elements are renamed by the script.

### Clean-up DEMO model



Figure 12: script: "Clean up DEMO model"

Select the package that needs to be cleaned up. With right mouse click, Specialize and Scripts you can select the "Clean up DEMO model". A system output window opens and shows the steps performed up to and including "Finished clean up for package <package name>".

This version also creates the package for elements where the package is not yet present. And for the Action Rules, he looks at the Alias and takes the first two positions for the transaction type number. In AM, he creates an ARS package and a TKxx package for each transaction type, where xx stands for the relevant transaction type number. Within each TKxx package, he creates two packages: Initiator and Executor if the name of the AR is also provided with the relevant coordination act. Depending on who is responsible for the ARS in question, it will be placed in that package.

Image with text

Auto-generated description

Figure 13: result clean-up ARS'n

### Update ARS

The third and final script adds a composite diagram of the ARS type for all ARn present on the TPD and places the event, assess and response part on it and puts the standard text in English of these ARS types in the notes. All texts can then be adjusted manually if necessary.

Since important parameters of these texts are based on the model, a good naming convention is very important. Of course, this convention adheres to the basics as stated in DEMOSL (version: 4.7.2). In a subsequent version of the MDG, these will also be used to check for modelling rules.

The time to run the script is when all ARs have been placed on the TPD of TKxx.

Naming convention:

Transaction Kind Name, name van TK: [casekind] <production -ing verb>

Transaction Kind Number, alias van TK: <xx>

TAR(xx) Name: [casekind] <production nominal>

Executing Actor Role: [casekind] <production nominal>

Product Kind Name< name van PK: the [casekind] is <production past tense>

Action Rule Name: contains the coordination act

Action Rule Alias: xx.ii where ii is the ARS-type number with:

01: response to request, request other TKs

02: response to request, respond with decline or promise

03: response to promise, request other TKs

04: response to promise, respond with execute and declare

05: response to declare, respond with accept or reject

06: response to revoke request, respond with allow or refuse

07: response to revoke promise, respond with allow or refuse

08: response to revoke declare, respond with allow or refuse

09: response to revoke accept, respond with allow or refuse

10: response to request selfstarter, respond with decline or promise

11: response to promise selfstarter, respond with request other TKs

12: response to promise selfstarter, respond with execute and declare

13: response to declare selfstarter, respond with accept or reject

By using the patterns from the toolbox, you automatically get the correct ARS types on your TPD. So with a normal transaction type: 1 to 9. In the case of a self-starting transaction type 6 to 13.

There is also the possibility to indicate an ARS type 14, but this is an empty shell and the text is placed here manually and

## Value Types

DEMO assumes an attribute type when it can be expressed in a value type, such as the Standard Index for units. The list as defined in the last DEMO-SL is included in the MDG. These coding datatypes are called "SI (Value Types)". Importing the MDG will set this up automatically. If that is not the case, you can define once on an OFD or an HD diagram. With a right mouse click on the diagram itself, select "properties". Select "Elements" and set the dropdown box from "language" to "SI (Value Types)".

Image with text, screen recording, software, screen

Auto-generated description

Figure 14: Set City Index for Value Types

## Importeren predefined tagged values

The shapescripts use predefined tagged values that are not automatically included in the MDG. Go to Settings, Import Reference Data. Select the file "DEMO4\_MDG\_tagged\_values" in the <DEMO4 downloads> location.

Image with text, electronics, screen recording, software

Auto-generated description

Figure 15: Importeren reference data: tagged values

## Operation DEMO4 MDG

### Introduction

The operation of the DEMO4 MDG is set up per aspect model, per diagram type. The aspect models treated in order are:

1. Cooperation Model (CM Cooperation Model)
2. Process Model (PM procesmodel)
3. Fact Model (FM feitenmodel)
4. Action Model

Within the aspect model, a diagram type HD (Auxiliary diagram) is included. This is not an official diagram type of DEMO4. However, it gives the possibility to visually create the objects and establish the relationships that follow from the other aspect models.

In a yet-to-be-released Addon, this part will be taken over in integrated scripts, eliminating the need for additional (double) modeling. The aspect models are then synchronized after each change on an aspect model.

### Cooperation Model (CM Cooperation Model)

In the right place in the project browser, select a package and with a right mouse click and select "Add diagram...". Select the profile "DEMO4\_MDG\_Diagram" and the diagram type "CSD".

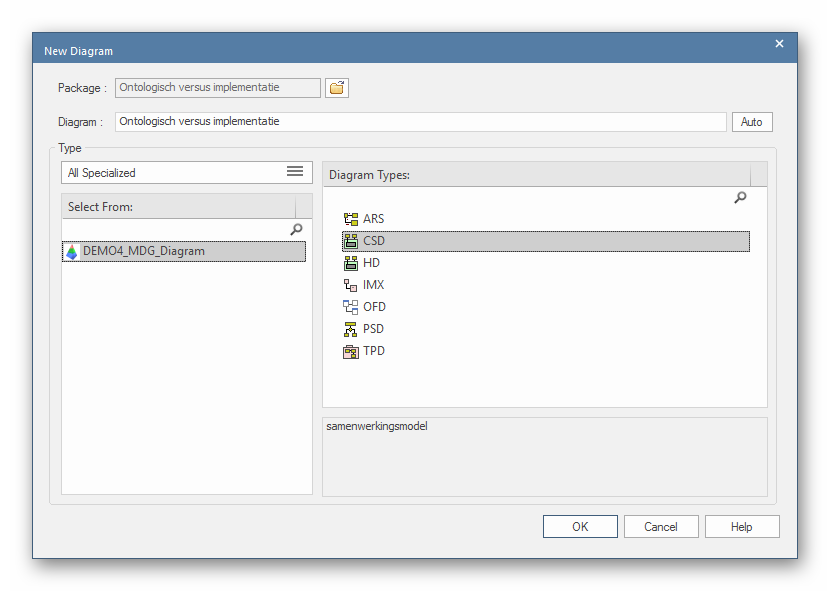


Figure 16: Add diagram CSD

Check that the correct toolbox is available:

Image with text

Auto-generated description

Figure 17: Toolbox CSD

The example below shows all the possibilities of a CSD. You'll see a CTAR, multiple TARs, a self-starting TAR, and an MTK. Between a CTAR and a TAR an initiator link, between a TAR and a TAR an initiator link, between a self-starting TAR and a TAR an initiator link. An access link between a TAR and an MTK. A wait link between a TAR and a TAR.

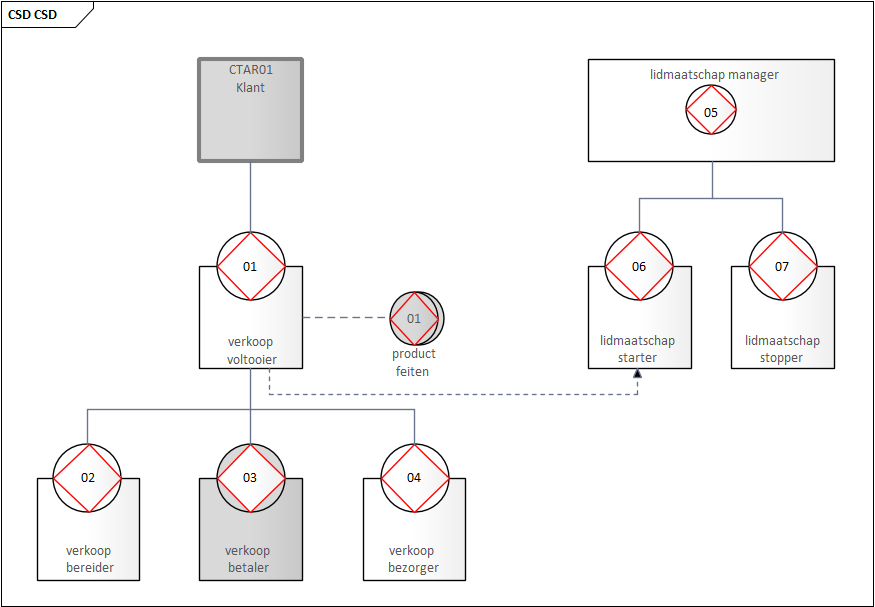


Figure 18: example CSD

You can make the associations by selecting a link in the toolbox and making an association from source symbol to target symbol.

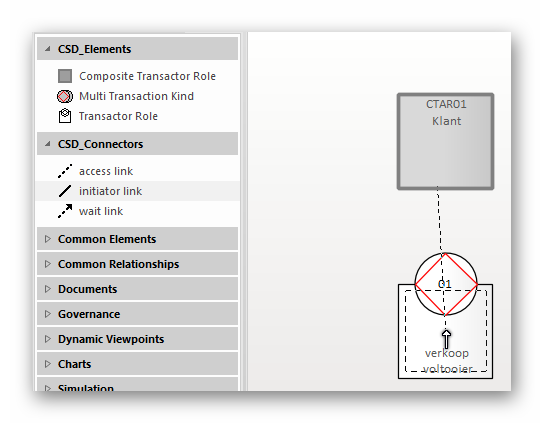


Figure 19: Association using the toolbox

For all links, you can select a source symbol and the arrow next to the symbol. By drawing that link to the target symbol, you will be offered all possible associations. In the example below "TAR/AR or CTAR initiates TAR/TK".

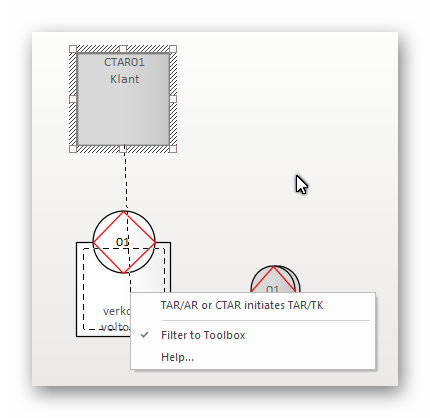
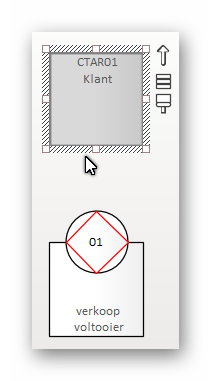


Figure : Association using quicklink

The same diagram type can be used to create a TPT with a custom table. You can find these in the Dynamic Viewpoints in your toolbox. With a right mouse click you can select "Edit custom table".

Image with table

Auto-generated description

Figure 21: example TPT as a custom table

You can do the same for a BCT and a CUT. In this MDG, this is still manual work, where you have to keep track of whether a change in the model affects the various components. In a future Addon, this will be synchronized with modeling rules across the entire model.

When creating a new Transactor role, a composite diagram of type HD is automatically created. This is not an official DEMO diagram type, but it is useful for creating all the DEMO3 elements and associations that automatically result from the existence of the TAR. This is also automated in the Addon, so that it takes place underwater. This diagram also gives the possibility to make a DEMO3 OCD.

Image with text

Auto-generated description

Figure 22: composite diagram HD

After adding the properties of the TAR, it is advisable to include the composite diagram TAR<No>, so that you know that this diagram contains the elements and associations that follow from the existence of the TAR.

On the HD diagram you create the corresponding TK, AR, HP and ET with their underlying relationships.

Image with diagram, screenshot, text, line

Auto-generated description

Figure 23: Auxiliary diagram for the objects and relationships that follow from the TAR

The TPT shows that the TAR leads to the existence of the corresponding Transaction Kind, Executing Actor Role and Product Kind. The Entity Type based on the case child of the product child also exists and is created in the Transaction Kind, given as a parameter or made available externally. This is reflected in the Create Use Table. This diagram type can also be used to create the DEMO 3 view.

In the same ARS, the clause may state that an actor role is linked to a specific person:

**the** payer **from** [sale] **is a** [person].

Image with Text, Screen Capture, Font, Line

Auto-generated descriptionImage with text, diagram, line, screenshot

Auto-generated description

Figure 24: possibility to define the moment that a person should be linked to an actor role

The Transaction Kind also has a composite diagram for the TPD.

Image with text

Auto-generated description

Figure 25: composite diagram TPT

It is advisable to give the name of the composite diagram the TK<no>. In the accompanying toolbox there is a pattern that creates the TPD pattern on the diagram.

Image with Text, Screen Recording, Font, Number

Auto-generated description

Figure 26: TPD pattern for a normal transaction type and for a self-starter

All elements of the TPD are created with the TKXX parameter, the XX of which stands for the number of the transaction child. A script can be started that replaces this parameter for all elements with the correct number and also lists the Action Rules in the alias this number.

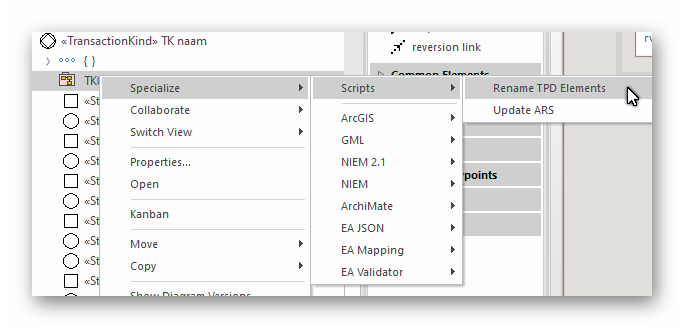


Figure 27: Rename TPD Elements

To do this, select the TPD diagram in the project browser. With a right mouse click select "Specialize", "Scripts" and the script "Rename TPD Elements". This script opens a popup window that asks for the TK number. Enter that and the script will rename all TPD elements.

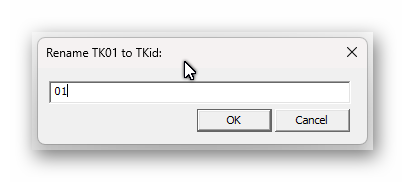


Figure : TK<No> imports

Image with diagram, Rectangle, screenshot, number

Auto-generated description

Figure 29: the rename script also includes the Action Rules in terms of alias number

### Process Model (PM procesmodel)

In the right place in the project browser, select a package and with a right mouse click and select "Add diagram...". Select the profile "DEMO4\_MDG\_Diagram" and the diagram type "PSD".

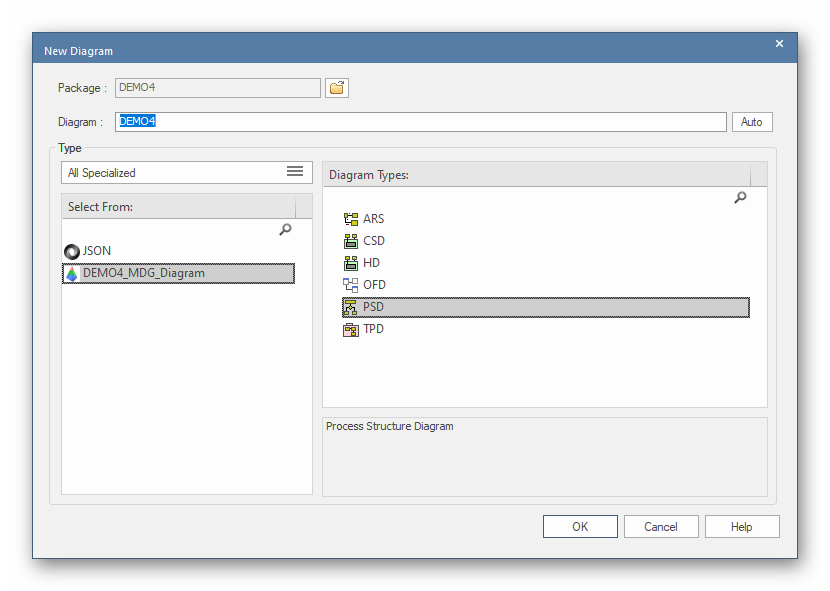


Figure 30: Add diagram PSD

The process model consists of a compact part and a "worst" part where the response and wait relationships are made between the acts and facts of transaction type steps. The compact part is a matter of dragging the existing objects onto the canvas. They have already been made at the CSD. The transaction types can be dragged onto the canvas in the form of a sausage that can be adjusted in width. By dragging the relevant transaction type steps on the canvas and placing them on the edge of the sausage, the correct response and wait links can be placed. The executor act also shows the product child. In that case, the product does not have to be dragged onto the canvas.

Image with text, diagram, screenshot, line

Auto-generated description

Figure 31: PSD compact and "sausages"

### Fact Model (FM feitenmodel)

In the fact model, all relevant Entity Types with Attribute Types and their interrelationships are included.

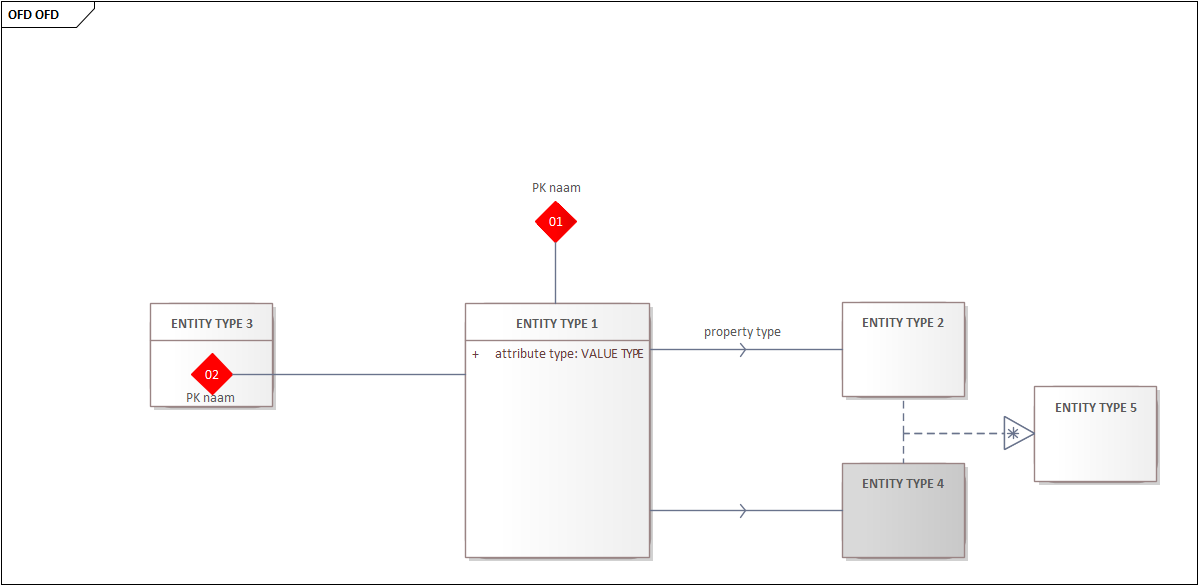


Figure 32:OFD

### Action Model

Not official according to the DEMO Specification Language is the possibility to indicate on the TPD where an Action Rule is used. The Action Rule Specification Diagram is a composite diagram of this.

If we pull out the TPD, we can turn it into a kind of timeline.

Image with Text, Screen Capture, Line, Font

Auto-generated description

Figure 33: TPD in the form of a timeline

The timeline shown is only applicable to the sequence in which coordination and production actions can take place. Also the revocations (revokes) can only take place from the moment that the fact to be revoked actually exists. If we look at the ARS'n, there is a clear difference between the regular transaction type and the self-starting transaction type.

Image with Text, Diagram, Plan, Screen Recording

Auto-generated description

Figure 34: TPD "normal" transaction type

Image with Text, Diagram, Plan, Technical Drawing

Auto-generated description

Figure 35: TPD Self-Starting Transaction Type

On the TPD, an AR looks like this (not an official DEMO symbol)

Image with text, clock

Auto-generated description

Figure 36: Action Rule 01

When creating an AR, a composite diagram is created for the ARS. If you use a pattern to create a TPD with the basic ARs placed on it, these charts will not be created yet. If you right-click on the TPD in the project browser and start the script "update ARS", these diagrams are created, the ARS elements are placed on them and the notes are filled in based on the model and the correct ARS type.

Image with Text, Screen Recording, Font, Number

Auto-generated description

Figure 37: ARS toolbox

It is therefore possible to build a TPD manually, place ARs, fill in the ARS. The patterns make it possible to create a large part of it automatically and to fill the notes correctly with the script for the ARS.

Finally, there is a script that can clean up the project browser and create the right packages. To do this, right-click on the highest package of the DEMO model in question, then "Specialize", "Scripts" and "Clean up DEMO model".

Image with text, line, screenshot

Auto-generated description

Figure 38: Clean up DEMO model

This script gives the modeler the possibility to completely ignore the location where elements are placed when creating the diagrams. By the time the modeling is complete, launch this script and the project browser will be cleaned up as configured in the script.

After running the script with a right mouse click on view of the relevant DEMO model, the result is:

Image with Text, Screen Capture, Font, Rectangle

Auto-generated description

Everything has been moved within the packages CM (cooperation model), PM (process model), FM (fact model) and AM (action model)

When we open the CM package we see:

Image with Text, Electronics, Screen Recording, Font

Auto-generated description

Figure 39: project browser layout CM

CTAR for all composite actor roles

HD for all unofficial help diagrams

MTK for all multi transaction kinds

A package for tables. In addition to the TPT, a CUT or a BCT can also be included as a table. TAR for all transactor roles

Image with Text, Screen Recording, Font, Number

Auto-generated description

Figure 40: project browser layout PM

AR for all actor roles

TK for all transaction kinds

And one or more process structure diagrams

Underneath a transaction kind is a composite transaction pattern diagram with a full TPD and below all transaction kind kind acts and facts.

Image with text, document

Auto-generated description

Figure : project browser format TK

Image with Text, Screen Capture, Number, Font

Auto-generated description

Figure 42: project layout browser FM

The fact model contains an ET package with all entity types with composite all associated attribute types.

PK contains all product kinds.

And there's an Object Fact Diagram

Image with Text, Screen Capture, Font, Document

Auto-generated description

Figure 43: project browser AM layout

The script divides the action model according to the types of transactions present. Each transaction type is given a package called TK<nr>, where nr stands for the transaction type number. Depending on the action rules present and the responsible actor role, the Initiator and Executor packages are also created, in which these action rules are placed. The script does this using the action rule name if there is a step kind act in it. The first two positions of the alias are used to determine the transaction kind. So you can agree on a possible naming convention on this.

An ARS diagram is composite to an action rule. There are three parts to this diagram: event, assess and response.

Finally, there is a package EL where all the existence laws are placed.

### Implementation matrix (IMX)

In this version 1.3 the possibility to relate the ontological model of DEMO to the different implementations modeled in ArchiMate is supported. So far, it has been limited to this. If there is enough interest, it can also be extended to other languages.

To model the implementation, the new IMX diagram and an implementation matrix can be used.

The diagram allows you to link all deployment objects with ArchiMate objects. In the matrix, you can link Transaction Kinds with Business Processes and Product Kinds with Business Products. A Transaction Kind Step Kind Act can also be linked to a Business Process.

This MDG version does not yet have a matrix as a template, but it will be available in the near future.

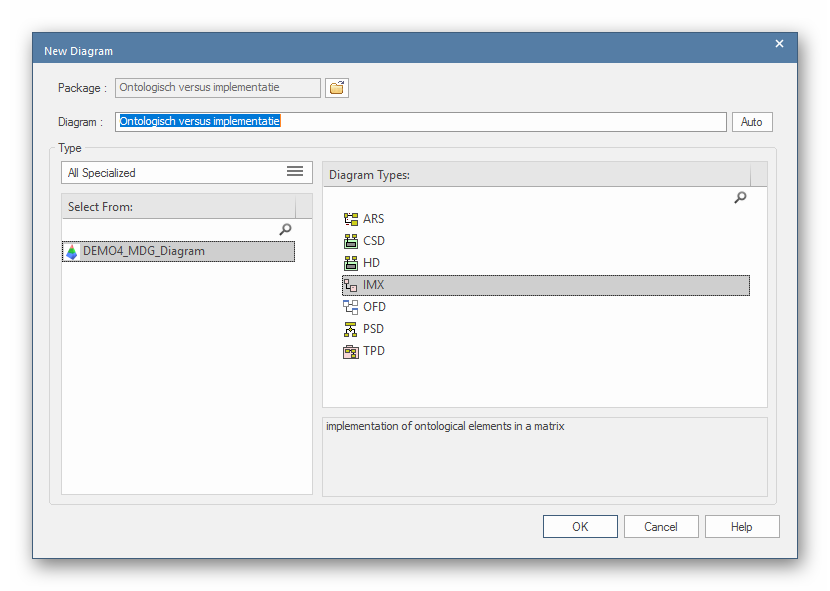


Figure 44: IMX available as a diagram

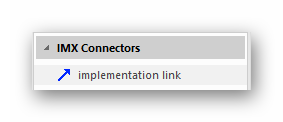


Figure : implementation link in de toolbox

Image with diagram

Auto-generated description

Figure 46: Business Process related to Transaction Kind

Image with diagram

Auto-generated description

Figure 47: Business Process related to Transaction Kind Step Kind Act

## Elaboration of Social Housing as an example

### Introduction

In the book Enterprise Ontology, section 12.4 describes the method of building an essential model.

### Transaction Product Table

Image with Text, Number, Font, Line

Auto-generated description

Figure 1: Social Housing Transaction Product Table

### Coordination Structure Diagram

Image with text, diagram, screenshot

Auto-generated description

Figure 2: Social Housing Coordination Structure Diagram

### Auxiliary diagrams (OER rounds)

Although not an official DEMO diagram, this diagram serves two purposes. The most important thing is that the analyst/modeler's train of thought can be captured in the model. In the OER method, the lines in the TPT are worked out line by line. You can put these steps in an auxiliary model. The second goal is that the elements from the toolbox of this diagram allow you to create a DEMO3 diagram.

Image with text, diagram, screenshot, line

Auto-generated description

Figure 3; Social Housing: Round 1 Help Diagram

First of all, the three elements that show the existence of TAR01, with production starting and casekind [registration] are TK01, registration starting for transaction kind 01. The initiator actor is CTAR01, (prospective) member and the executor is AR01, registration starter. The first one is created on the CSD and can be copied from the project browser. The second one is part of TAR01, but you need to create it here. In a future MDG version, we could also automate this, but going through the train of thought as an analyst is important. The third and final element of the TPT line is the product type "[registration] is started" which means that an instance of the [registration] entity starts to exist. Since the "starting day {DAY}" is an attribute type of the specialization [registration], namely [started registration], the product type gets a group link to [registration] and an entity "started registration" around it with the attribute type starting day.

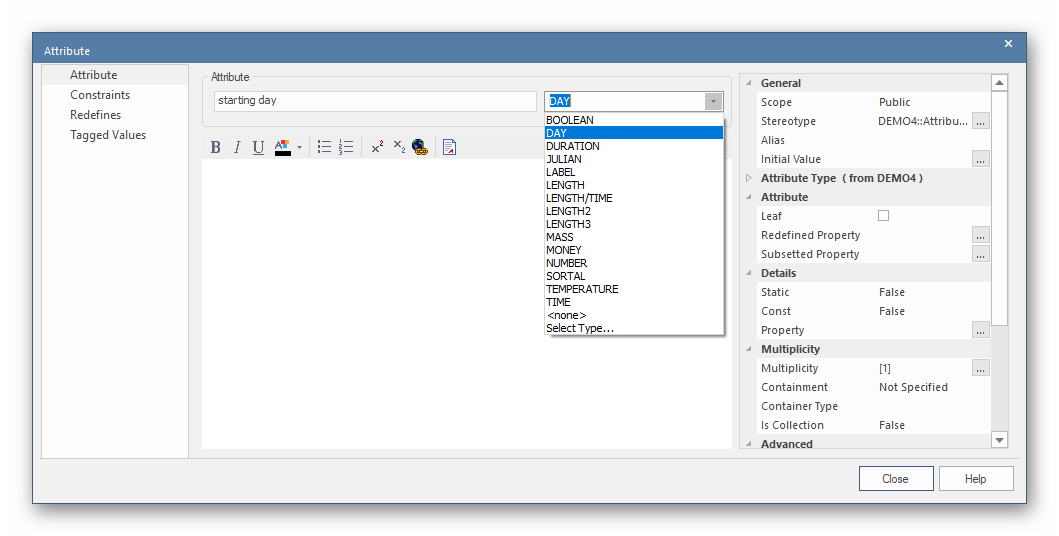


Figure 4: creation of attribute type with corresponding value type

By selecting an attribute type from the toolbox and placing it on the entity, you get an input screen where you can enter the name of the attribute type and a list of value types that were imported during the installation.

Image with Text, Screen Capture, Screen, Rectangle

Auto-generated description

Figure 5: Attribute Input Screen

If this is not already installed properly, you will get the default list. By going to the properties of the diagram, you can select the SI (value types) under Elements.

Image with text, screen recording, software, screen

Auto-generated description

Figure 6: Select SI (value types)

At the entity level and at the attribute level, you can indicate that it arises in this transaction type, by means of a create use link. Initially, transfer it from the entity to the transaction type. In the properties you can indicate what applies here:

C: create

D: derived

G: given externally

P: given as a parameter

With a left mouse click on the create use link, you can use Link to element Feature to select that it is an attribute and select the one in question.

Image with text, screen recording, screen, software

Auto-generated description

Figure 7: Link to element Feature

You can also indicate a property link between [registration] and [person] that it arises here. The property on the left in figure 3 are: "the member of [registration] is [person]" and "the payer of [registration] is [person]".

Image with text, screenshot, diagram

Auto-generated description

Figure 8: Auxiliary diagram belonging to round 2

Image with text, diagram, screenshot, line

Auto-generated description

Figure 9: Auxiliary diagram for round 3

Image with Text, Screen Capture, Diagram, Parallel

Auto-generated description

Figure 10: Auxiliary diagram belonging to round 4

Image with Text, Diagram, Plan, Technical Drawing

Auto-generated description

Figure 11: TPD belonging to TK01

Before we can use the ARS script, we need to clarify which ARs deviate from the standard. Of course, it is also possible to include the standard, but it does not provide any additional information. When we select a pattern from the toolbox, all relevant ARs are placed on the TPD by default. The Update ARS script hangs a composite ARS diagram and the corresponding ARS template underneath and fills the elements with the standard texts.

The cleanup script takes into account the naming convention and places the elements in the right place in the project browser. Not only that, but it's also useful to keep track of the details in the model.

Image with Text, Diagram, Plan, Technical Drawing

Auto-generated description

Figure : TPD belonging to TK02

Image with Text, Diagram, Plan, Technical Drawing

Auto-generated description

Figure : TPD belonging to TK03

Image with Text, Diagram, Plan, Technical Drawing

Auto-generated description

Figure 14: TPD belonging to TK04

### Roces Model

In the process model, within a transaction type tree, it is indicated how transaction types mutually influence each other. At the transaction type, step, child, response links and wait links are placed. In a Process Structure Diagram, the analyst / modeler can put this information. This is reflected in the ARS'n.

Image with screenshot, text, diagram, line

Auto-generated description

Figure : PSD 01: CTAR01 - TK01 - TK02

Image with diagram, screenshot, line, text

Auto-generated description

Figure : PSD 02: TK04 - TK02

Image with diagram, text, screenshot, line

Auto-generated description

Figure : PSD 03: TK04 - TK03

### ARS 01.2

Image with Text, Screen Recording, Parallel, Number

Auto-generated description

Figure : complete ARS 01.2

### ARS 01.3

Image with text, screenshot, number, software

Auto-generated description

Figure : complete ARS 01.3

### ARS 01.4

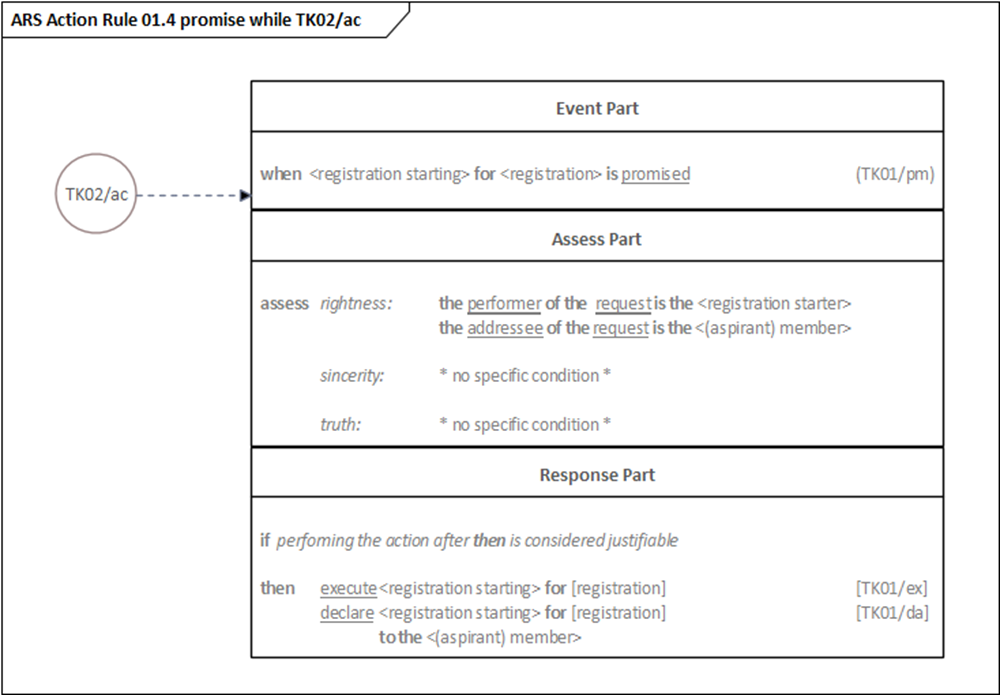


Figure : complete ARS 01.4

### ARS 02.5

Image with Text, Screen Capture, Number, Font

Auto-generated description

Figure : complete ARS 02.5

### ARS 03.2

Image with Text, Screen Capture, Number, Font

Auto-generated description

Figure : complete ARS 03.2

### ARS 03.4

Image with Text, Screen Recording, Font, Number

Auto-generated description

Figure ; complete ARS 03.4

### ARS 03.5

Image with Text, Screen Capture, Number, Font

Auto-generated description

Figure : complete ARS 03.5

### ARS 04.10

Image with Text, Screen Recording, Number, Parallel

Auto-generated description

Figure : complete ARS 04.10

### ARS 04.11

Image with Text, Screen Recording, Number, Parallel

Auto-generated description

Figure : complete ARS 04.11

### ARS 04.12

Image with Text, Screen Capture, Number, Font

Auto-generated description

Figure : complete ARS 04.12

### ARS 04.13

Image with Text, Screen Recording, Font, Number

Auto-generated description

Figure 25: complete ARS 04.13

### FM complete

If we now bring the results of all rounds together in the fact model, we have the end result FM.

Image with Text, Screen Capture, Diagram, Font

Auto-generated description

Figure : Fact Model Complete