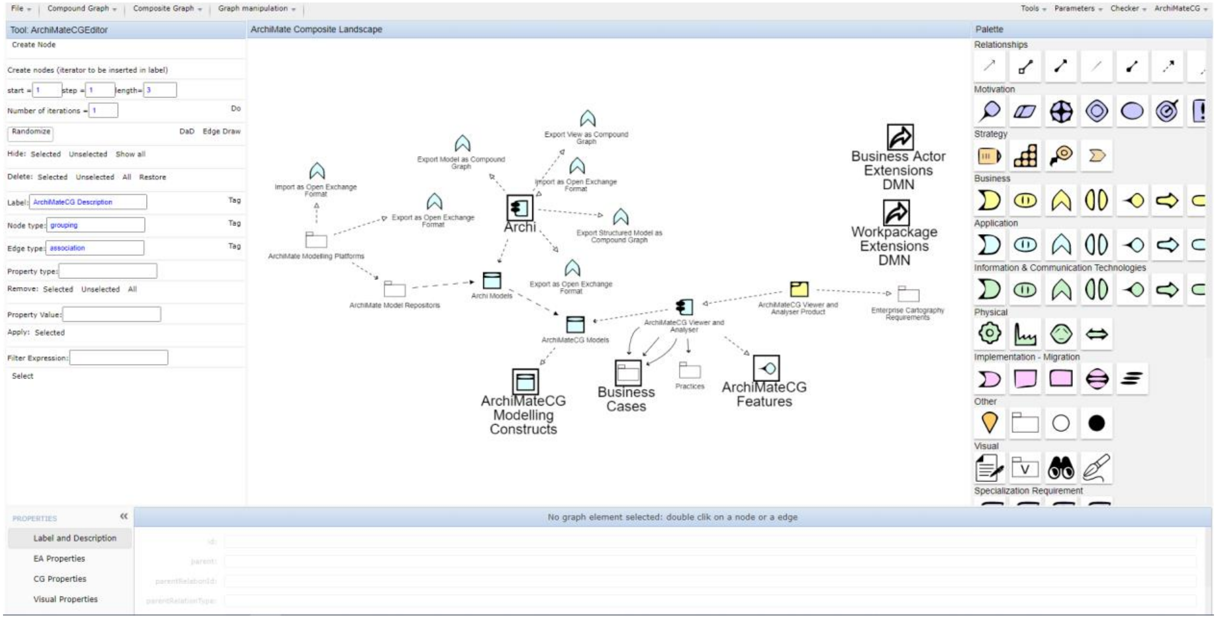
ArchiMateCG

ArchiMate Compound Graphs Viewer and Analyser


File
Compound Graph
Composite Graph
Graph manipulation
Tools
Parameters
Checker
ArchiMateCG
Tool: ArchiMateCGEditor
Create Node
ArchiMate Composite Landscape
Palette
Relationships
Motivation
Strategy
Business
Application
Information & Communication Technologies
Physical
Implementation - Migration
Other
Visual
Specialization Requirement
Randomize
DaD
Edge Draw
Do
Tag
Export View as Compound Graph
Export Model as Compound Graph
Import as Open Exchange Format
Import as Open Exchange Format
Export as Open Exchange Format
Archi
Export Structured Model as Compound Graph
ArchiMate Modelling Platforms
Export as Open Exchange Format
Archi Models
ArchiMate Model Repositories
ArchiMateCG Models
ArchiMateCG Viewer and Analyser
ArchiMateCG Viewer and Analyser Product
Enterprise Cartography Requirements
Business Actor Extensions DMN
Workpackage Extensions DMN
ArchiMateCG Modelling Constructs
Business Cases
Practices
ArchiMateCG Features
PROPERTIES
Label and Description
EA Properties
CG Properties
Visual Properties
id:
parent:
parentRelationId:
parentRelationType:

Create nodes (iterator to be inserted in label)

start = 1 step = 1 length= 3

Number of iterations = 1

Hide: Selected Unselected Show all

Delete: Selected Unselected All Restore

Label: ArchiMateCG Description

Node type: grouping

Edge type: association

Property type:

Remove: Selected Unselected All

Property Value:

Apply: Selected

Filter Expression:

Select

No graph element selected: double clik on a node or a edge

User Guide

Version 1.0.20

Table of content

Table of figures

Introduction

ArchiMateCG is a solution aiming at assessing the value of using ArchiMate Interactive Compound Graphs for the various stakeholders involved in architecting the enterprise for its digitalisation and who adopted the ArchiMate® 3.1 language as the standardised architecture description language.

ArchiMateCG aims at serving actors dealing with End to End Digital Processes, Model Based System Engineering, Product Lifecycle Management industrial approach (as defined by CIMDATA) and Interoperability of Enterprise Application.

For this, **ArchiMateCG** supports creation of advanced interactive compound graphs visualisation and algorithms in order to analyse architecture descriptions and blueprints produced using the ArchiMate language.

ArchiMateCG is not intended to replace legacy ArchiMate modelling platforms and enterprise repositories, but to complement and extend them with previously described features.

Consequently **ArchiMateCG** comes with various import and export functionalities, and complementary sets of scripts for **Archi**® and **Enterprise Architect**®

ArchiMateCG is realised by combined usage of standardised Web technologies such as HTML, SVG and JavaScript, in order to run on and require only a Web Navigator.

ArchiMateCG is also a research incubator, aiming at demonstrating and assessing research results on Continuous Operational Interoperability for digital collaboration within and between enterprises applying Model Based approaches. It will in future versions address the combined usage of Linked Data, Semantic Web, Standardised Application Protocols and Graph based technologies.

ArchiMateCG is currently a prototype constituting a Proof of Concept and a Proof of value concerning innovative combination of ArchiMate modelling, Compound Graph visualisation and analysis and Semantic Web technologies.

Thanks to:

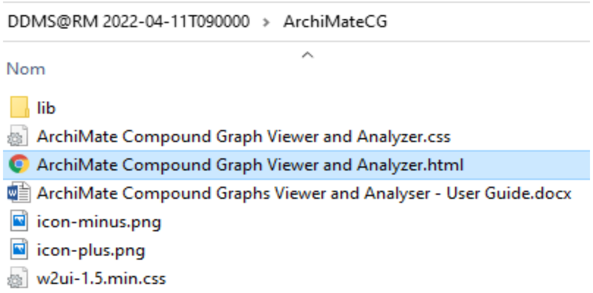
- Brice Fayolle, Celine Boudou, Matthieu Ladjyn who provided feedbacks in terms of value and potential usage
- Thierry Cordier who provided many specifications of use cases with value creation for solution architects and specification of export scripts from Enterprise Architects in order to support these use cases
- Phil Beauvoir and Jean-Baptiste Sarrodie, the authors of **Archi**® which is an actual source of inspiration in terms of ArchiMate modelling, and of models by relying on the associated jArchi scripting plugin as easy way for data exchange between **Archi**® and **ArchiMateCG**
- Max Franz and the development team of Cytoscape.js who have been producing an incredible library in terms of advanced visualisation and processing of interactive compound graphs. Cytoscape.js has been heavily used as an enabling library for ArchiMateCG.

Installing and Launching ArchiMateCG

ArchiMate download can be done:

1. Downloading a zip archive on a dedicated Airbus Google (access request to be address to Nicolas Figay – Nicolas.figay@airbus.com)
2. Through access to a dedicated GitHub server (access request to be addressed to Thierry Cordier – thierry.cordier@airbus.com)

The zip archive files are of the following form: *client* aaa-mm-ddThhmmss.zip
with *client* being the targeted client/team for who a configuration[1] of ArchiMateCG was delivered. Figure 1 illustrates it with the release published the 11th of March 2022 at 9 o'clock for the DDMS@RM team.



DDMS@RM 2022-04-11T090000 › ArchiMateCG

Nom

- lib
- ArchiMate Compound Graph Viewer and Analyzer.css
- ArchiMate Compound Graph Viewer and Analyzer.html
- ArchiMate Compound Graphs Viewer and Analyser - User Guide.docx
- icon-minus.png
- icon-plus.png
- w2ui-1.5.min.css

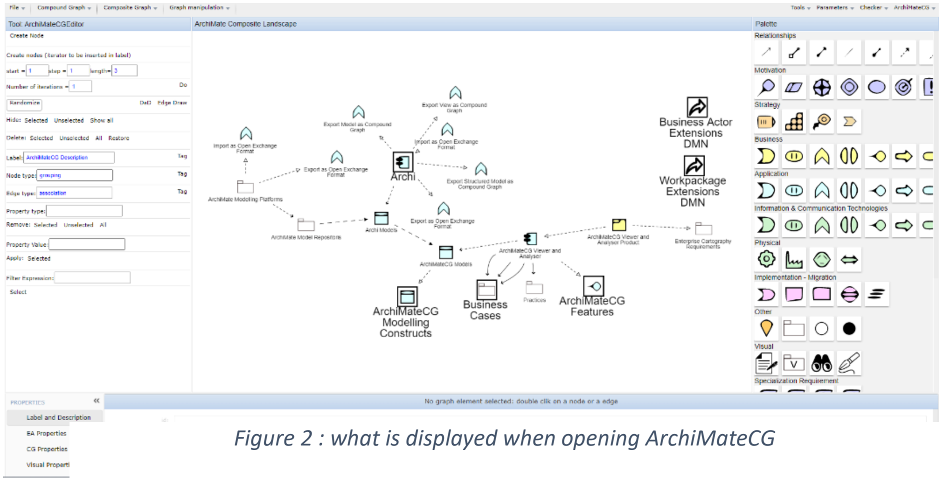
Figure 1: HTML file to open for launching ArchiMateCG

The files are to be installed in a given target root repository. The application is launched by opening the "**ArchiMate Compound Graph Viewer and Analyzer.html**" with Web browser (a recent version of Google Chrome is preferred, but should work as well with Safari, Internet Explorer, Microsoft Edge or Firefox as soon as supporting properly JavaScript and SVG – no compatibility matrix realised at this time.)

Launching ArchiMateCG

When opened, the working environment interface appears with as default open graph the one representing the ArchiMateCG architecture as an interactive compound graph ready to be navigated.

Note: the initial graph to be open can be changed. It is defined in /lib/initialgraph.js



Tool: ArchiMateCGEditor

Create Node

Create nodes (iterator to be inserted in label)

start = 1 step = 1 length= 3

Number of iterations = 1

Label: ArchiMateCG Description

Node type: grouping

Edge type: association

No graph element selected: double clik on a node or a edge

File
Compound Graph
Composite Graph
Graph manipulation
Tools
Parameters
Checker
ArchiMateCG
Do
Randomize
DaG
Edge Draw
Hide: Selected Unselected Show all
Delete: Selected Unselected All Restore
Tag
Property type:
Remove: Selected Unselected All
Property Value:
Apply: Selected
Filter Expression:
Select
ArchiMate Composite Landscape
Import as Open Exchange Format
Export Model as Compound Graph
Export View as Compound Graph
Import as Open Exchange Format
Export as Open Exchange Format
Archi
Export Structured Model as Compound Graph
ArchiMate Modeling Platforms
Export as Open Exchange Format
ArchiMate Model Repositories
Archi Models
ArchiMateCG Models
ArchiMateCG Viewer and Analyser
ArchiMateCG Viewer and Analyser Product
Enterprise Cartography Requirements
ArchiMateCG Modelling Constructs
Business Cases
Practices
ArchiMateCG Features
Business Actor Extensions DMN
Workpackage Extensions DMN
Palette
Relationships
Motivation
Strategy
Business
Application
Information & Communication Technologies
Physical
Implementation - Migration
Other
Visual
Specialization Requirement
PROPERTIES
Label and Description
EA Properties
CG Properties
Visual Properti

Figure 2 : what is displayed when opening ArchiMateCG

[1] The delivered packages may encompass core Cytoscape with or without some of the described tools, or with very specific tools responding to some specific needs of targeted users according to their profile (which kind of stakeholders they are, and the intended usage of ArchiMateCG). Let's note that ArchiMateCG has been developed an agile and continuous way in collaboration with many communities, being projects or organization, with as an objective to make emerge best practices relying on continuous innovation.

The working environment is divided into the following sections:

- **The graph visualisation panel**. By default, it is positioned on the centre and named “Cytoscape”.
- **The Tool panel**. It is positioned on the left. By default, it contains the ArchiMateCGEditor tool panel, which provides buttons and parameters in dedicated fields for acting on the graph for various editing actions such as creating, hiding or removing graph elements, tagging nodes or edges with labels or ArchiMateCG types, searching and selecting nodes by applying a filter expression. It also contains selection widgets for activating Randomization of automated compound graph layout, Drag and Drop (Button “DaD”) for grouping nodes for containments (compound nodes) and for drawing edges (Button Edge Draw). The Toolbar of the panel changes when selecting another tool in the Tools menu, such as ArchiMateCG Editor (default), Coloured Map, Viewpoints, Colouring, Animated Demonstration and ACG Analytics. The functionalities made available through these toolbars are described later in the guideline.
- **The palette panel**. It is positioned on the right side, and proposes a set of icons grouped per ArchiMate layers for ArchiMate modelling constructs, enriched by some ArchiMateCG complementary modelling constructs icons specific to ArchiMateCG for reflecting structural representation of the physical (folders, packages) and logical (views, viewpoints) ArchiMate model structures. It also includes sets of specialisation for some given ArchiMate modelling constructs, relying on ArchiMate specifications for extensibility based on specialisations. The palette can be filtered by applying globally a viewpoint (cf. Tools). Using Shift DblClick, it allows to filter visibility of the different kind of nodes or edges.
- **The properties panel**. By default, it is positioned at the bottom. Different folders are proposed to group the displayed properties: label and Description, EA properties, CG Properties, Visual Properties.

The different panels can be rearranged by dragging their frontiers in order to change the sizes of the panels.

In addition, a menu is proposed on top of the screen allowing you to launch various commands.

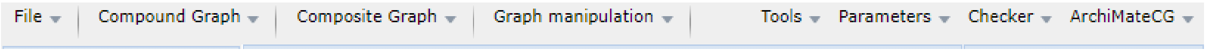


Figure 3: the main ArchiMateCG menu

Working with ArchiMateCG

Interacting with the graph[2]

The graph is the main windows of ArchiMateCG.

- Using the mouse, you can change the position of nodes clicking on them, maintaining the mouse down and then dragging the node where you want. Let's relax the mouse button (mouse up) at the place you want the node to stay.
- By clicking on a compound node (with black square around the icon), the node is greyed and a "+" appears. By clicking on the node, you expand the node, with an automated layout applied for positioning the nodes using the algorithm of the layout (by default FCOSE Cytoscape extension layout). When expanded and selected, a "-"appears on the compound node. By clicking on it, you collapse the node and an automated layout is applied.
- When double clicking on the node or on an edge, you select it and you make appear on the left pane the graph element label and type (Node type and Edge type in two different fields). Clicking on the "tag" button at the right of these fields, you will apply the label or the type to the selected elements on the graph.
- Selection is made by clicking an element on the graph pane with the mouse or by selecting a region with the mouse. You can also, with the shift key down, select or unselect several nodes or regions without deselecting what is already selected.
- The delete, hide or remove operations which can be applied from left panel buttons or from the menu "Graph Manipulation", can be applied on selected, unselected or on all the graph elements appearing on the graph.
- When double clicking on an icon of the palette, the node type field or the edge type field on the left panel are taking the type value associated with the clicked icon. It is then possible to perform type tagging on a selection, or to create a node clicking on the "Create Node" button. A new graph node is then created with the label and Node type indicated on the left pan, which is automatically positioned by the current automated layout. If no node or several nodes are selected, the node will be created without a parent. If a node is selected, the created node will be a child of the selected node.
- When double clicking on a graph element, the properties of the bottom pan are updated with the values of the graph element, and the title of this pane indicated what is selected indicated the type of the node or edge, the id of the graph element and if specialised, the specialisation of the node or edge. Data properties of the graph element can be changed on the properties panel and applied to the selected element of the graph. They can also be changed or added from the left panel, providing content for Property and Value fields, and clicking the button "Selected" on the left of Apply.
- When clicking on "Edge Draw" button, you change the drawing mode. Nodes can't be moved anymore with the mouse, but clicking on a node, it is now possible if not relaxing the mouse button to draw an edge, with the global edge type indicated in the field Edge type, to a target node you have to go over and then relaxing the mouse button. Then, if the ArchiMate relationships rule are relaxed, the relation is created. If not relaxed but enforced, the relation will be created only if the ArchiMate relationships is allowed between a source and a target having such types (what is allowed is specified within the relations table provided in the ArchiMate 3.1 specifications. Indicating if enforced or relaxed can be made through the Parameters>>ArchiMate Relationships Rule menu, by selecting Enforce or Relax. By default, rules are enforced. Enforcement doesn't prevent not allowed relationships to be part of the graph, only their creation by "Edge Draw" mode. Note it is possible to identify not allowed relationship by using the Checker (Checker menu)

[2] Pay attention: if most of the actions are undoable since Version 1.0.15, some of them can't be undone.

Creating a new graph

From the default opened graph, you can remove all the graph elements and start creating a new graph relying on previously described node creation or typed edge drawing.

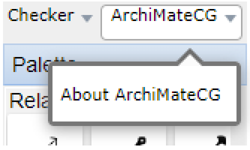
You can also apply various operations made available from the menu.

You can also load or import a graph, and you will be asked if you want to replace the already existing graph or to add the loaded/imported graph to the existing one. If some elements already exist (same id), the properties will be updated with those coming from the loading/import.

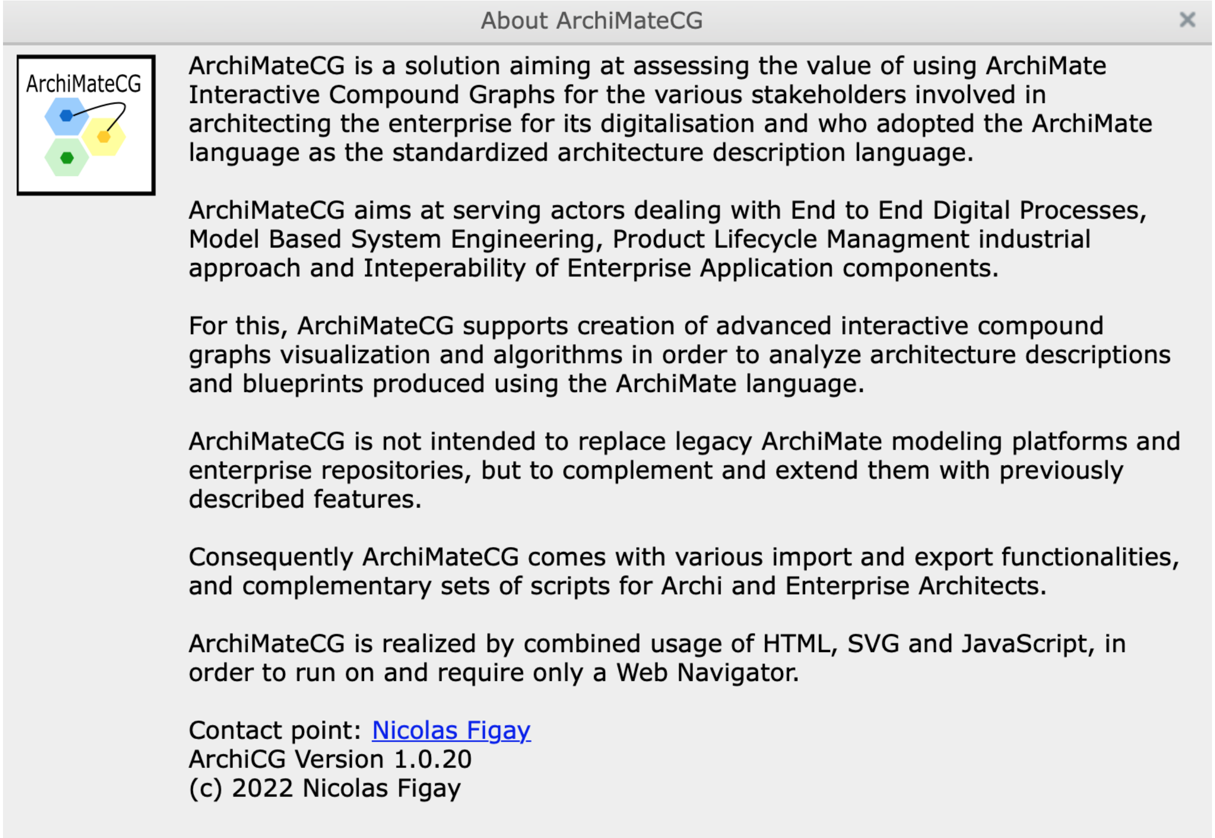
Operations available through the menu

ArchiMateCG

About ArchiMateCG



Clicking on this menu display a short presentation of ArchiMateCG.



About ArchiMateCG

ArchiMateCG is a solution aiming at assessing the value of using ArchiMate Interactive Compound Graphs for the various stakeholders involved in architecting the enterprise for its digitalisation and who adopted the ArchiMate language as the standardized architecture description language.

ArchiMateCG aims at serving actors dealing with End to End Digital Processes, Model Based System Engineering, Product Lifecycle Managment industrial approach and Inteperability of Enterprise Application components.

For this, ArchiMateCG supports creation of advanced interactive compound graphs visualization and algorithms in order to analyze architecture descriptions and blueprints produced using the ArchiMate language.

ArchiMateCG is not intended to replace legacy ArchiMate modeling platforms and enterprise repositories, but to complement and extend them with previously described features.

Consequently ArchiMateCG comes with various import and export functionalities, and complementary sets of scripts for Archi and Enterprise Architects.

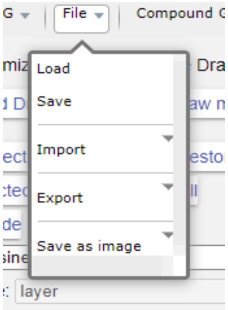
ArchiMateCG is realized by combined usage of HTML, SVG and JavaScript, in order to run on and require only a Web Navigator.

Contact point: Nicolas Figay
ArchiCG Version 1.0.20
(c) 2022 Nicolas Figay

Figure 4: About ArchiMateCG

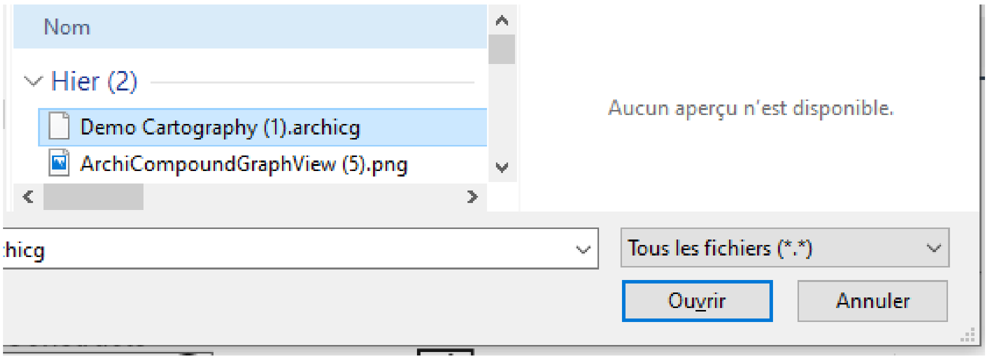
File

This menu allows to load or save ArchiMateCG Compound graphs, to import or export graphs on various format or to save graphs as images

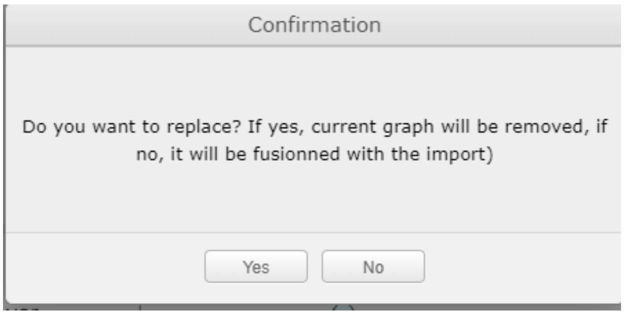


Load

This command opens a file selector, where you can open an ArchiMateCG file (extension .archicg).


Nom
Hier (2)
Demo Cartography (1).archicg
ArchiCompoundGraphView (5).png
Aucun aperçu n'est disponible.
hicg
Tous les fichiers (*.*)
Ouvrir
Annuler

When a valid file is selected, a popup asks for confirmation of replacing the current graph, or to merge it with the loaded one.


Confirmation

Do you want to replace? If yes, current graph will be removed, if no, it will be fusionned with the import)

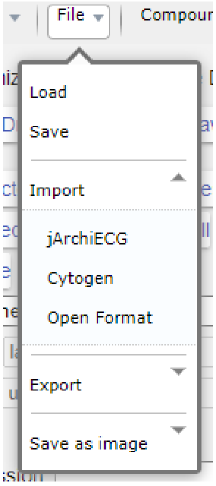
While opened, the graph is displayed.

Save

Clicking on this menu opens a popup window proposing the name “myGraph” with extension “.archicg”. The name can be changed. Then the file is downloaded on the download folder of the navigator. Note that as the application, which is run on a Web navigator, doesn't access the file management system of the hosting machine, it is not possible to select a targeted folder. Files are consequently to be put on dedicated directories manually.

Import

Import allows importing files with various provenances.



jArchiCG

These are files with the JSON format defined by Cytoscape for graph data, adapted for proposing ArchiMate/ArchiMateCG types for nodes or edges, and eventually object properties for compound and composite nodes. It corresponds to what is produced with a dedicated jArchi scripts to be used with Archi, for exporting various contents from it which are dedicated to ArchiMateCG. The content includes some data specific to Archi, such as folders, drawings, (name, value) properties without types and extensions.

Since version 1.0.13, most of the usual errors encountered when loading these files are captured and not blocking anymore. The list of encountered errors are displayed after the logging. When related to missing source or target node, a blank node is created, i.e. a node with type="blank-node" and with the id indicated for the source or for the target. If importing a new file which will be added to the current graph, the blank node is changed with addition of the node data. Let's note that trying to import a graph with already existing elements which are not "blank-nodes" in the current graph will let them unchanged.

Cytogen

These are files with the same JSON format as jArchiCG, extracted from an Enterprise Architect export script, aiming at analysing architecture of solutions. The content includes some structural information, such as packages, modules, etc.

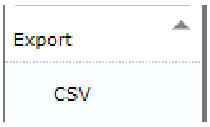
Open Format

These are XML files structured according to the schemas defined by the Open Group Open Exchange File format for ArchiMate, supported for import and import by many products.

Export

Export to various format and exchange protocols (currently only CSV files)

CSV



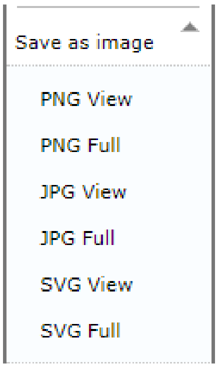
Clicking on this menu allows to export the content of the current selection, as 3 zipped files, one for the nodes, one for the relations and one for the properties. Ids are used for links between these objects, those initially imported, or those created with ArchiMateCG (based on UUIDs).

OWL

Clicking on this menu allows to export the content of the current selection as an OWL file with the JSON LD syntax. It indicated the import of an OWL file derived from the ArchiMate Language extended with ArchiMateCG concepts and eventually used specialization and data/object properties if willing to use them with reasoners. All properties not defined as properties in this file are included as annotations. More details on intended usage with Protégé are providing latter in a dedicated section.

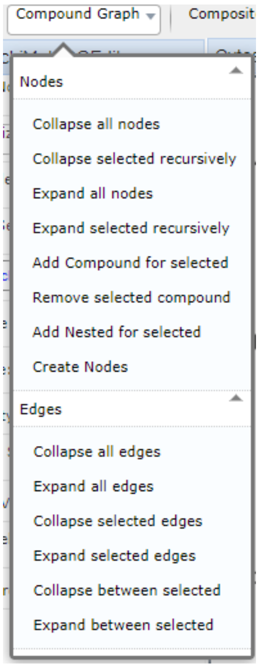
Save as image

Through this menu, it is possible to save the current graph as an image, what is on the pane (view) or the whole graph (full) if the pane shows only a part of the graph. PNG, JPG and SVG formats are supported. The image is downloaded in the download folder of the navigator.



Compound Graph

This menu gives access to operations related to manipulation of compound graphs, with ability to collapse or expand nodes and graphs, and eventually to create compound nodes around a set of selected nodes.



Collapse all nodes

All the nodes are collapsed, and only the nodes without parents and their relationships, including those between their child nodes. The layout is reorganised.

Collapse selected recursively

Selected nodes are collapsed. The layout is reorganised.

Expand all nodes

All the nodes of the graph are expanded. The layout is reorganised.

Expand selected recursively

All the selected nodes are expanded recursively. The layout is reorganised.

Add Compound for selected

For a given selection of nodes, if they are at the same level, a new compound node is created which is the parent of all selected nodes.

Remove selected compound

For a selected compound node, it is removed and the child nodes become child nodes of the parent of the parent, or nodes without parent if the selected node is a root node.

Add nested for selected

For given selected nodes a containing node is created which becomes the parent of these nodes, and is a child of the parent of the selected nodes. It works only if the selected nodes are all siblings.

Create Nodes

It allows you to launch node creation through a popping window and propose, unlike the "Create Node" button of the ArchiMateCG toolbar, entering some properties for the node creation and creating node, with the ability to reuse value of the properties of the created node for a next creation. The idea is to propose an alternative way for node creation when willing also to create properties.

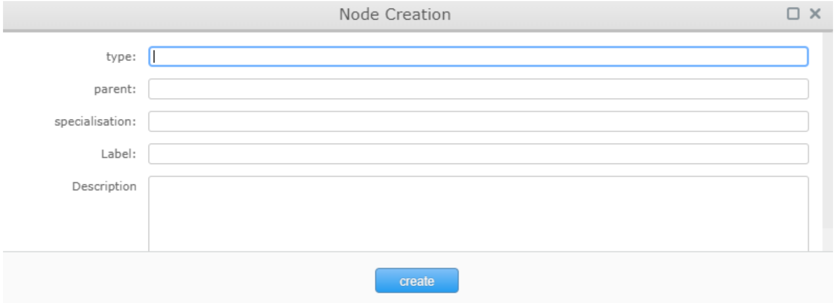


Figure 5: Node Creation Form

The nodes are created all to the same fixed position, and it is needed to relaunch a layout after the creation of the nodes for having them positioned accurately.

Collapse all edges

All the edges between two nodes are replaced by a single edge, with indication of the number of edges it replaces.

Expand all edges

All the collapsed edges are expanded.

Collapse selected edges

All the selected edges are collapsed when relevant, i.e. edges between the two same nodes in the group of selected edges.

Collapse between selected

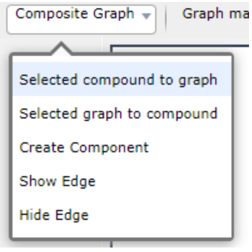
The collapsed edges between selected nodes are expanded.

Expand between selected

The edges between selected nodes are collapsed.

Composite graphs

A composite graph is a graph containing nodes with composition relationships. It can be displayed as a graph, or a compound node. This menu provides operations allowing to switch from one visual representation to the other, showing or hiding edges for compound graphs. It is also possible to create a component for a given node, i.e. a child node with a composition relationship between the parent node and the child node.



Selected compound to graph

Considering the selected graph, all the composition relationships displayed as compound nodes are displayed as graph

Selected graph to compound

Considering the select graph, all the composition relationships displayed as graphs are displayed as compound nodes.

Create Component

Considering a selected node, a new child node is created plus a composition relationship between the selected parent node and the child node.

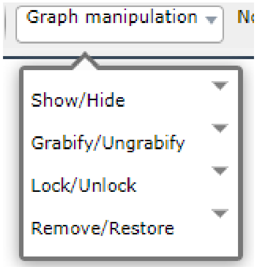
Show edge

Considering the selection, all the composition edges for which the composition source node is the parent of the composition target node are made visible.

Hide edge

Considering the selection, all the composition edges for which the composition source node is the parent of the composition target node are hidden.

Graph manipulation



This menu gives access to operations concerning the ability or not to see, grab or change node positions for what is selected or what is not selected. It also allows removing graph elements, and eventually to restore what was removed during a working session.

To grab consists in moving nodes using the mouse (click on a node and with mouse down, move the node with your mouse).

Lock nodes can't be graphs or move programmatically (so they are not affected by the automated layout).

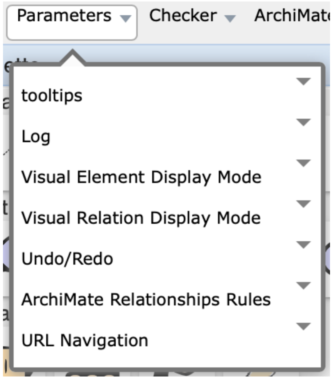
Collapsed nodes or nodes without parent are prior to considering the positioning and compared to compound expanded nodes, so moving them will affect positioning of expanded parent nodes containing them.

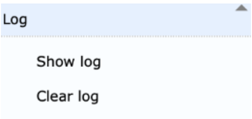
Restore and show all will concern all what was hidden or removed during the session, except removed compound nodes (menu Compound Graph>> Remove Selected compound).

Remove actions are not undoable!

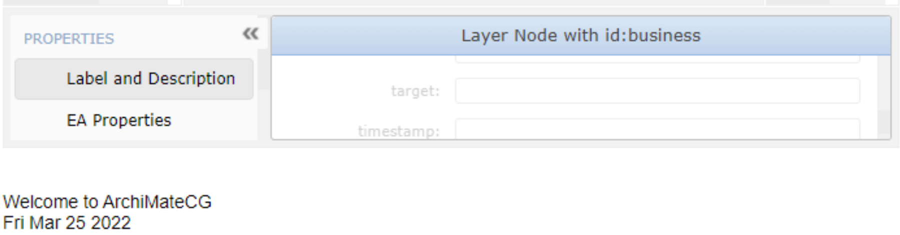
Parameters

It provides for parameterizing tooltips, Log, Visual Mode and Undo/Redo


Parameters
Checker
ArchiMat
tooltips
Log
Visual Element Display Mode
Visual Relation Display Mode
Undo/Redo
ArchiMate Relationships Rules
URL Navigation

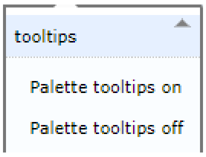


Log: This menu allows to show or to clear logs of ArchiMateCG, which was created programmatically as Web navigators in enterprises may be subject to log deactivation. So this basic window, below the Properties pane, provides a solution for being able to inform the user that something went wrong. When launched, ArchiMateCG sends a welcome message on the log (cf. Figure 6). So log can be shown (Show log) and cleared (Clear log).

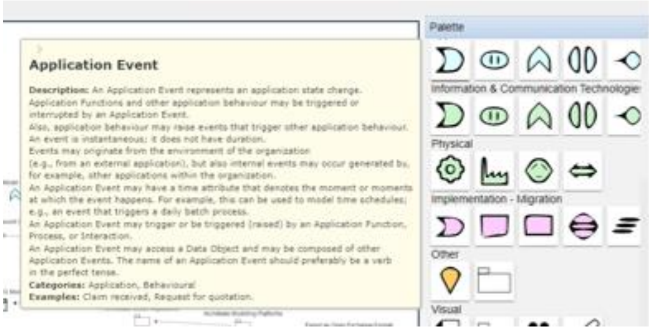


```
Welcome to ArchiMateCG
Fri Mar 25 2022
```

Figure 6: Log display with Log set to on



Tooltips: This menus allows to activate or to deactivate tooltips on various interface objects. For the current version, only the palette is supported. User Interface and graph will follow.



Application Event

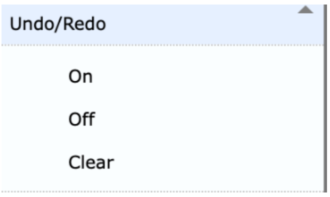
Description: An Application Event represents an application state change. Application Functions and other application behaviour may be triggered or interrupted by an Application Event.
Also, application behaviour may raise events that trigger other application behaviour. An event is instantaneous: it does not have duration.
Events may originate from the environment of the organization (e.g., from an external application), but also internal events may occur generated by, for example, other applications within the organization.
An Application Event may have a time attribute that denotes the moment or moments at which the event happens. For example, this can be used to model time schedules; e.g., an event that triggers a daily batch process.
An Application Event may trigger or be triggered (raised) by an Application Function, Process, or Interaction.
An Application Event may access a Data Object and may be composed of other Application Events. The name of an Application Event should preferably be a verb in the perfect tense.
Categories: Application, Behavioural
Examples: Claim received, Request for quotation.

Palette
Information & Communication Technologie
Physical
Implementation - Migration
Other
Visual

When clicking on “Palette tooltips on”, the definition of the concept represented by an icon is shown when mouse is over the icon, and hidden when mouse is not over anymore. The provided definition is the one given by ArchiMate specifications when it is an ArchiMate language concept. When clicking no “Palette tooltips off”, only the name appears.

Visual Relation Display mode (available since V1.0.16): This menu allows to select between Edges and Nodes. When Edges (default), all ArchiMate relationships are defined as an edge, with the proper rendered symbol defined by ArchiMate. When Nodes is selected, the edge is hidden and a node is created with the relationship symbol, plus two edges from this not and linking the source and the target of the relationship. This alternative representation is not specified by ArchiMate. It aims at making more explicit the underlying data model and to show that an ArchiMate relationship, as well as an ArchiMateCG edge, is in fact a typed object which can have data properties, and referencing two nodes, one being the source and the other being the target. Such a mode is proposed in order to prepare a future functionality concerning the ability to create and visualize object properties, i.e. typed references between ArchiMate model elements or relationships. Displaying object properties as arcs from the property owner to the reference object will be possible when activating the “Nodes” visual relation display mode.

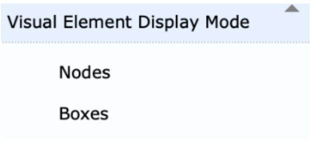
Undo/Redo (available since V 1.0.15): This menu proposes to set ability to Undo/Redo to “On” or to “Off”. When “On”, it is possible to undo (Ctrl X) or redo (Ctrl Y) many (but not all) actions performed on the graph: moving, creating, deleting, hiding a graph element, changing a label (with Tag buttons), and rearranging a graph (collapse, expand). The menu also propose “Clear”, which consists in deleting the do/undo stack (so no more redo/undo done before clearing is available anymore).



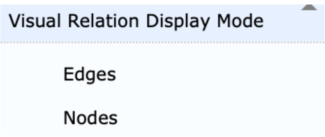
The functionality is particularly helpful if willing to correct a mistake when tagging or deleting nodes

Let’s note that undo/redo for graph rearrangement is not working very well, and that creation of several nodes in one time (with Do button of the ArchiMateCGEditor tool) is split in several operations for the do/undo, and not in the single operation. So the functionality is subject to improvement and extension of the considered actions which are subject to do/undo.

Visual Element Display Mode: This menu proposes “Nodes” visual mode or “Boxes” visual model. With “Nodes” mode, each node is associated an icon and a label displayed under the icon. With “Boxes” mode, each node is associated a box containing the label and the icon at the top left side of the box, as displayed on diagrams proposed by several modelling tools such as Archi. The “Boxes” mode is related to the export of a current graph as view, as the size of the boxes will be preserved for produced diagrams for Archi or ArchiMateCG.

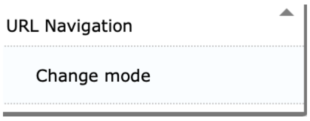


At this stage, an issue exists when words in the label or when the label are too long. Then the label is going out of the box. This is due to the currently considered features of the used libraries. Usually, a label should not be too long for readability. Consequently, solving the issue is not currently considered as a priority. Also, addressing it automatically for each model element could be quite resource *consuming at it should be addressed each time rendering a node. Let's note that when exporting the view as diagram, the label remains usually in the box for ArchiMateCG, and in Enterprise Architect*.

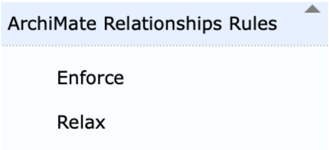


Visual Edge Display Mode: this mode allows to represent relationships as edges or as nodes (with two edges, "from" and "to"). Representing relationships as nodes is a way to highlight that a relationships is in fact an object with an identifier and properties, just as the model elements, and with two object properties, source and target.

We can imagine in future versions of ArchiMateCG to also include object properties and to visualize them, just as "source" and "target", or to include not only binary relationships, but also n-aries relationships with edges from the relationship to the other model elements with indication of their roles in the relationship.

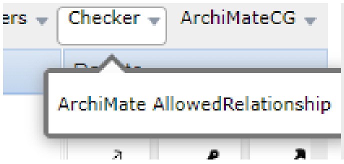


URL Navigation (available since V1.0.17): This allows to change mode related to being able by double clicking on a node to open the URL contains in the property "url" of the node. If the property doesn't exist, nothing happens. By default, it is not activated. Clicking on "Change mode" change the mode from "deactivated" to "activated" and from "activated" to "deactivated". It is possible to change the name of the property containing the URL using the Navigation tool.



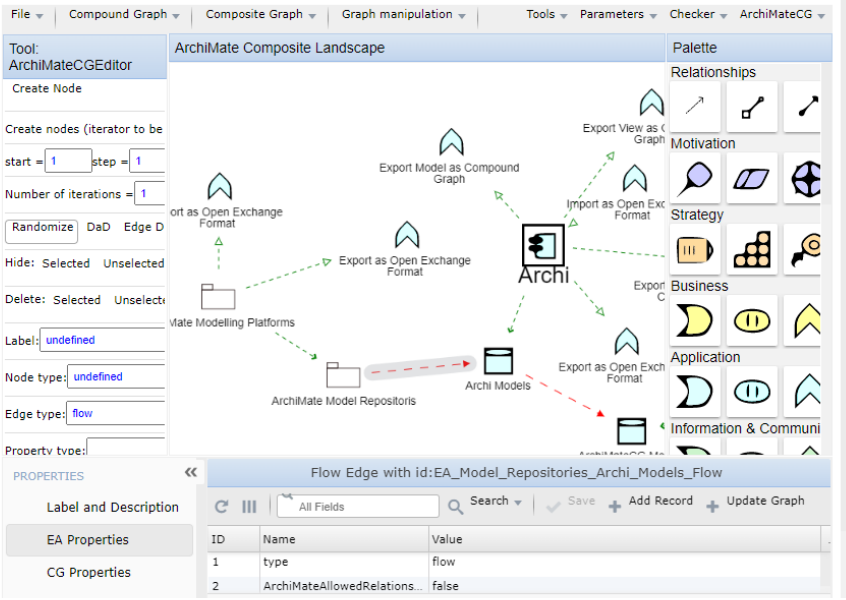
ArchiMate Relationships Rules (available since V1.0.16): this menu allows to indicate if the rules concerning ArchiMate Relationships are enforced or not when editing the model. It is completed by the checker, which will indicate the relationships which don't respect this rules by changing the colour (red) of the arcs representing them.

Checker



ArchiMate AllowedRelationship: clicking on this button creates when not existing and set the AllowedRelationship value to true or false. In addition, the colour of allowed relationships is changed to green, the colour of not allowed relationship is changed to red. Non ArchiMate relationships colour is unchanged. This applies only to visible relationships.

E.g. applying the checker to the initially loaded graph, we can see that some of the relationships it contains are not allowed by ArchiMate.



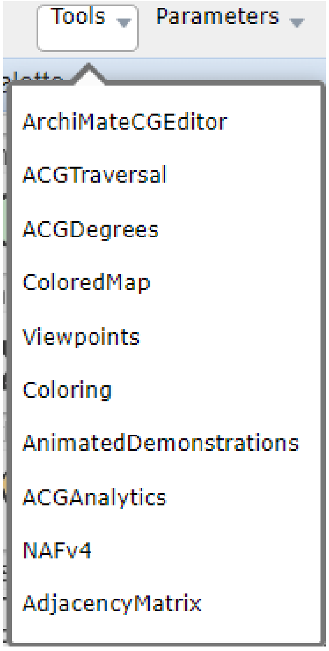
ID	Name	Value
1	type	flow
2	ArchiMateAllowedRelations...	false

Figure 7: Non valid ArchiMate relations put in red by the checker

Note: it is possible to filter (hide, remove) relying on the created property

It is planned in future version to extend rules which can be checked.

Tools



Tools is a feature supports a modular approach and extensibility for ArchiMateCG.

It comes with a specific toolbar which appears on the Left Panel and provides access to particular functionalities. The initial panel is also the first tool proposed in the tool list, **ArchiMateCGEditor**.

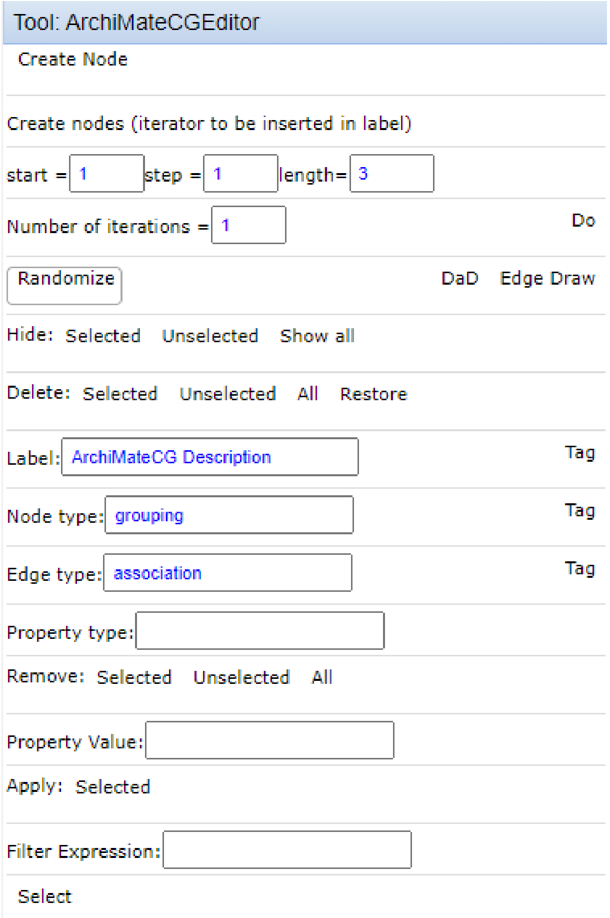
The ArchiMateCGEditor default tool panel provides access to many command allowing to quickly manipulate the graph, being for filtering or for edition, as reflected in the next figure.

The “Create Node” button creates a new node with Label and Node Type as defined in the panel. If no graph node selected or if several nodes are selected, the new created node is a top node. If one node is selected, the created node is a contained node of the selected node.

With the “Do” button, several nodes are created with the Node type indicated on the tool pan. The number of created nodes correspond to “Number of iterations”. The label is constituted of the type name, followed with “-“and with a string having the length indicated in the pan, and a number created from the start, field and number of iterations field. For each iteration, the number is start+step*currentIteration, prefixed with as many “0” as needed for reaching length.

E.g. with start =1, step=2, length=3, number of iterations=3 and Node Type =”grouping”, 3 nodes will be created having for type “grouping” and with labels “groupin-001”, “grouping-003” and “grouping-005”. This functionality is useful when willing to create several nodes from ArchiMateCG, completing a legacy graph or creating it from scratch.

Note: other ways to quickly generate a graph including both nodes and edges are under study, being from legacy data (e.g. using CVS data) or from graph generators (e.g. for data generator for testing).



Tool: ArchiMateCGEditor

Create nodes (iterator to be inserted in label)

start = 1 step = 1 length= 3

Number of iterations = 1

Label: ArchiMateCG Description

Node type: grouping

Edge type: association

Property type:

Property Value:

Filter Expression:

Create Node
Do
Randomize
DaD
Edge Draw
Hide: Selected Unselected Show all
Delete: Selected Unselected All Restore
Tag
Tag
Tag
Remove: Selected Unselected All
Apply: Selected
Select

“Randomize” button allows to indicate if collapse or expand actions are to rearrange randomly the graph when performing changes on the graph or not.

Hide and Delete can be performed on what is selected, on what is unselected or on the whole (visible) graph (same commands than then on the menu “graph manipulation”).

Label, Node type and Edge type are fields linked to global variables, which are used by Tag buttons (applied on current selection) and “Create Nodes”.

Property Type and Property Value fields can be used together for creating a property with the indicated type and value on the selected nodes, or to remove a given property on the selected, unselected of all nodes having the same property name than the one indicated on the field “Property Type”. (This operation is currently no subject to Undo/Redo).

The “Filter Expression” field allows to give an expression that will be used for selecting graph elements corresponding to this expression when clicking the associated “Select” button. The used language is the one coming Cytoscape.js, and is reflected on the section “Filtering with filter expression”

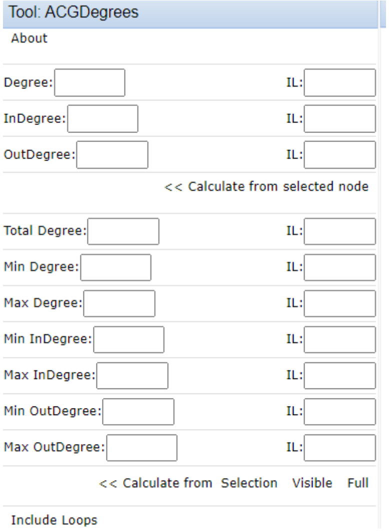
New extensions have been added which are described in the next pages.

ACGTraversal

This is a search toolbar, based on graph specific algorithm, allowing to search and select neighbourhood and shortest path (cf. “Searching with graph analytics” for detailed usage description).

ACGDegree

This is a tool allowing to calculate the degrees (as defined by graph theory) of a node or of a set of nodes. This also shows the built in functionalities that Cytoscape.js provides to ArchiMateCG. Degrees calculation is a feature which can be used for more sophisticated analysis.



The toolbar allows to calculate degrees for:

A single node. It provides degree, in degree out degree, including (IL) or not loops.

Let’s select a node and click on << Calculate from selected node.

For a selection, all the visible graph or the full graph, the calculated degrees, including loops (IL) or not, are total, min and max degrees, plus min and max in and out degrees.

ColoredMap

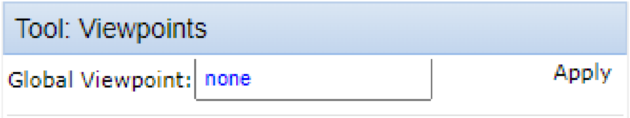
It now contains an improved version of what was initially proposed in the previous version. Cf.

Colourizing a graph according to the values of a given data property.

Viewpoints

It integrates the notion of viewpoint, as defined by ISO42010 and applied in ArchiMate, and is based on the viewpoints proposed in the ArchiMate 3.1 specification and implemented in Archi. Each viewpoint comes with targeted stakeholders, a purpose, a concern and a set of ArchiMate constructs to be used by the views built according to this viewpoint. When selecting one of the viewpoints, the palette is filtered, proposing only the constructs related to the viewpoint. In addition, the opacity of the elements typed with language constructs which don't belong to the viewpoint is changed by making them more transparent (similar to Archi, which uses the term "ghosted").

The selected viewpoint becomes a "global" viewpoint applied to the whole compound graph opened in ArchiMateCG. At this time, it is just a way to filter the model. It can be used for building views according to viewpoints on top of an ArchiMate model compound graph.



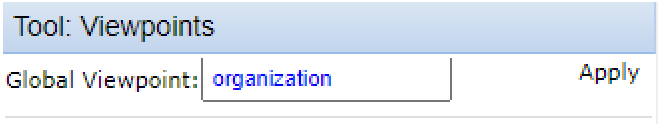
Tool: Viewpoints

Global Viewpoint: none

The toolbar proposes to enter the global viewpoint name, and then it can be applied by clicking on the Apply button.

There is list of viewpoints supported by the tool but not provided as a choice list (future feature). The list, which is aligned with the one of the specification and with Archi, is the following: "none", "application_cooperation", "application_structure", "application_usage", "business_process_cooperation", "capability", "goal_realization", "implementation_deployment", "implementation_migration", "information_structure", "layered", "migration", "motivation", "organization", "outcome realization", "physical", "product", "project", "requirements_realization", "resource", "service realization", "stakeholder", "strategy", "technology", "technology_usage" and "value_stream"];

When selecting a viewpoint and applying it, the palette is modified by proposing only the constructs related to the viewpoint.



Tool: Viewpoints

Global Viewpoint: organization

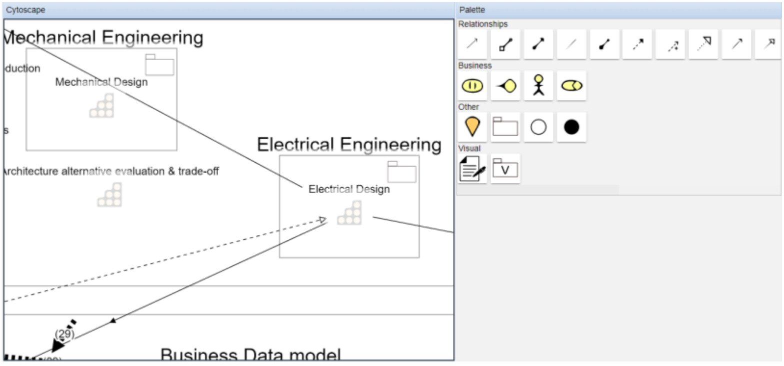


Figure 8: Palette filtered for organisation viewpoint and model element not belonging to the viewpoint ghosted

In addition, all the model elements which are not related to the viewpoints are not fully opaque (they are made transparent, like ghosts). It allows to filter a model, or to enforce when creating a graph to use only constructs allowed for a given viewpoint.

A dedicated library has been developed for the ArchiMate viewpoints, with for each viewpoint the name, the description, the purpose, the concern, the stakeholders and the allowed constructs. This was created from Archi, which is itself fully aligned with the ArchiMate Specifications.

It will be used for proposing new features with the next version of the tool, plus some demonstrations explaining usage and interest of viewpoints. It will also be used for exploring current limitations and potential high added value innovative extensions.

Save visible graph as view

It is possible to export, relying on the Open Exchange Format, the currently visible graph as a view with the selected global viewpoint. At this stage, the name of the exported model is “Test”.

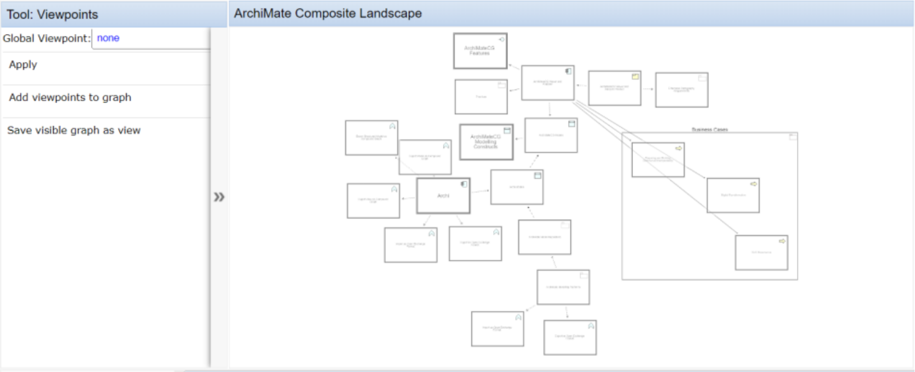


Figure 9: Illustration of visual mode put on boxes mode before Open Format export

There is a default size (width and height) with an applied scale depending on the current Visual Mode (Nodes or Boxes), which produce visual element of a uniform size on the exported view (diagram). What is mainly preserved includes locations of nodes, size of compound nodes and containment of the child nodes.

Note: Visual Mode can be changed through the Parameters menu.

If considering the previous figure, the export, after import in Archi, looks like in Figure 10, and like in Figure 11 after import in Sparx Enterprise Architect.

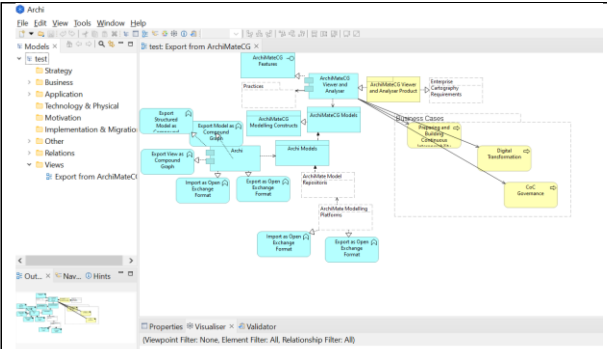

Archi
File Edit View Tools Window Help
Models
test
Strategy
Business
Application
Technology & Physical
Motivation
Implementation & Migration
Other
Relations
Views
Export from ArchiMateCG
test: Export from ArchiMateCG
Business Cases
Properties
Visualiser
Validator
(Viewpoint Filter: None, Element Filter: All, Relationship Filter: All)

Figure 10: The view as it appears after import in Archi

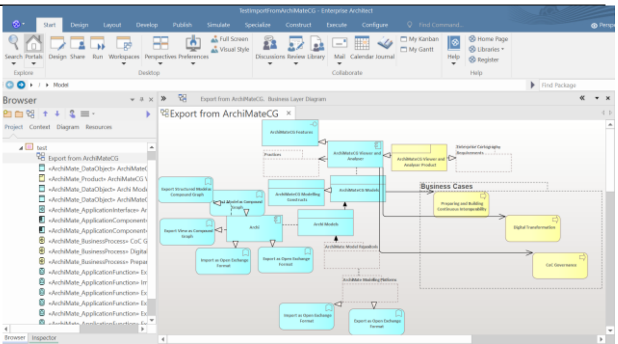


Figure 11: The view as it appears after import in Sparx Enterprise Architect

While the visual features provided by ArchiMateCG are not as rich as those provided for diagramming by usual modelling tools (e.g. Archi, Enterprise Architect, etc.), the main interest is to take advantage of filtering coupled with automated layout for compound graphs, which doesn’t have equivalent in the other tools. So it can help a lot for fast diagramming.

It is planned in future versions of the Viewpoints tool:

- *to attach them to view objects and to add complementary features related to viewpoints and providing value to the architects and other users of ArchiMateCG.*
- *to support definition of complementary viewpoints*
- *to support complementary visual features to be supported by ArchiMateCG and exported with “Save Visible Graph as View” (e.g. sizing of nodes as boxes, colors or edge styles).*
- *to support complementary exchange format dedicated to compound ArchiMate graphs extending the coverage of the Open Exchange Format.*

NAFV4 tool

This is a pre-alpha version of a tool which should deal with ArchiMate viewpoints as defined by the version 4 of the NATO Architectural Framework. In this last version, ArchiMate is one of the candidate language with the Unified Architecture Framework metamodel.

As the ArchiMate viewpoint are not yet specified, the tool only propose the current list of the NAF viewpoints, without any defined set of ArchiMate construct to be used for it.

So not useable at this stage, but more something to announce future support of NAFV4 viewpoints.

Eventually, some specific mapping could be proposed and implemented before the specifications to be published as an exercise for being able to assess these specifications. Let's follow.

Reference: "NATO Architecture Framework, Version 4", 31st of August 2022

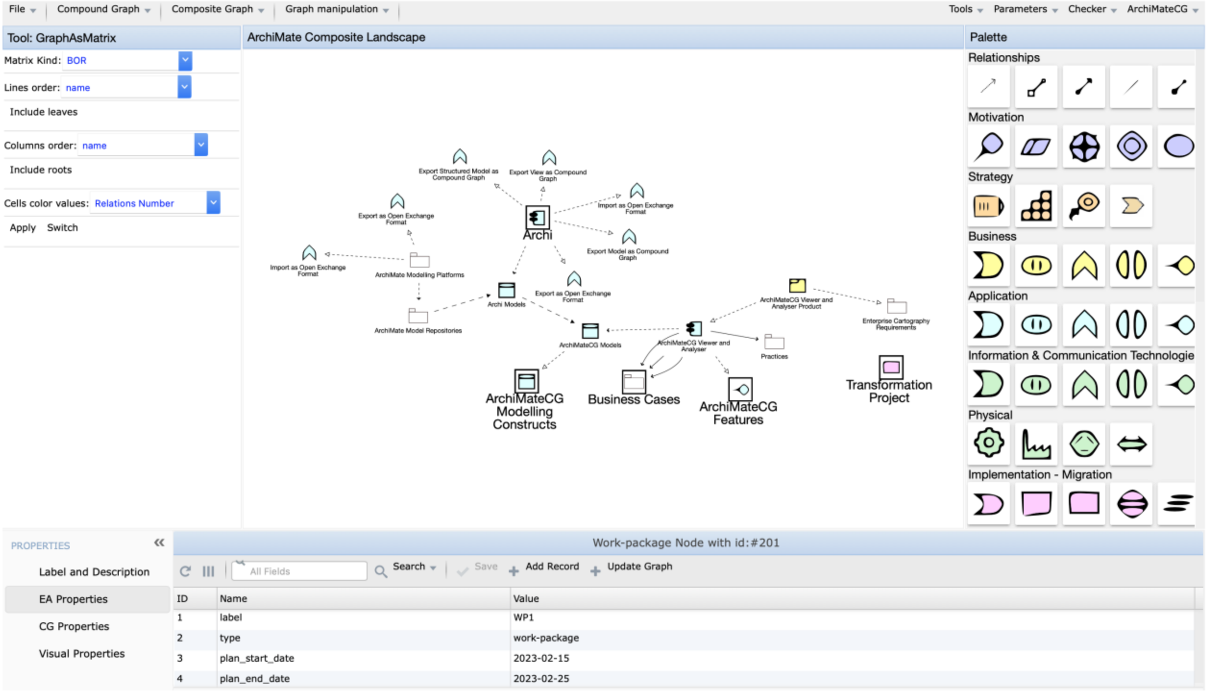
Graph as Matrix tool

In graph theory and computer science, an adjacency matrix is a square matrix used to represent a finite graph. The elements of the matrix indicate whether pairs of vertices are adjacent or not in the graph. This is an alternative way to visualize a graph, preventing the "hairball" views, i.e. too many arcs on a graph making it difficult to visualize a graph. It not accurate for identifying paths between nodes, a matrix makes it easier to make visible other kinds of complex information, in particular when coupled with advanced interactive dynamic visualization.

The proposed tool here replaces the Adjacency Matrix tool proposed in previous version, as a complete rethinking and new development. The rational and approach are explained in the chapter "The Matrixes in ArchiMateCG".

This is an alpha tool aiming at exploring how to make an accurate usage of such technics for supporting enterprise architects (and other stakeholders) activities.

When selecting it, the dedicated toolbar appears on the left pan, allowing to select the order which be used for the elements of the scales (mainly name at this stage). After selecting elements on the graph, it is then possible to click on the "Apply" button. A matrix will then be displayed on the central pan, replacing the graph. Clicking "Switch" allows to switch between graph and matrix representations.


File
Compound Graph
Composite Graph
Graph manipulation
Tools
Parameters
Checker
ArchiMateCG
ArchiMate Composite Landscape
Palette
Relationships
Motivation
Strategy
Business
Application
Information & Communication Technologie
Physical
Implementation - Migration
Export Structured Model as Compound Graph
Export View as Compound Graph
Export as Open Exchange Format
Import as Open Exchange Format
Archi
Export Model as Compound Graph
Import as Open Exchange Format
ArchiMate Modelling Platforms
Archi Models
Export as Open Exchange Format
ArchiMate Model Repositories
ArchiMateCG Models
ArchiMateCG Viewer and Analyser Product
Enterprise Cartography Requirements
ArchiMateCG Viewer and Analyser
Practices
ArchiMateCG Modelling Constructs
Business Cases
ArchiMateCG Features
Transformation Project
PROPERTIES
Label and Description
EA Properties
CG Properties
Visual Properties
All Fields
Search
Save
Add Record
Update Graph

Tool: GraphAsMatrix

Matrix Kind: BOR

Lines order: name

Include leaves

Columns order: name

Include roots

Cells color values: Relations Number

Apply Switch

Work-package Node with id:#201

ID	Name	Value
1	label	WP1
2	type	work-package
3	plan_start_date	2023-02-15
4	plan_end_date	2023-02-25

Figure 12: Tool Graph as Matrix toolbar

The button “Matrix Kind” aims at selecting the kind of matrix to be created. At this stage, only one is proposed, “BOR” (Binary Oriented Relations matrix). Others will be proposed in the next versions.

The button “Include leaves” when selected indicates that lines are created even for nodes without outgoing relations (could be of use for analysis, and for future functionalities related to create relations directly by interacting with the matrix).

The button “Include roots” when selected indicates that column are created even for nodes witout incoming relations (could be of use for analysis, and for future functionalities related to create relations directly by interacting with the matrix).

The buttons “Column order” and “Lines order” allow to select what defines the order of lines and columns: name, count or group.

- “Name” corresponds to the label of the node.
- “Count” corresponds to the number of outgoing relations for the source nodes (lines), to the number of incoming relations for the target nodes (columns)
- “Group” is using a property named “” for the nodes

The following figure shows the resulting matrix, each colored square indicating that a relation or severall exist between the model element at the row level (source) and the model element at the column level (target), and the reverse.

For the current version:

- The matrix considers displayed model elements (nodes) and oriented relationships(edges), with source nodes at left side and target at the top side.
- Different colors reflects different numbers of relation, but the legend is not provided. When the mouse is over a cell, the number of relationships is displayed, and the source and target labels are highlighted (headers of line and columm

- On the top left side of the pane of the matrix, the kind of displayed matrix plus information about the displayed matrix are displayed. The intentation is that it will be be part of futur exports as images or SVG, when the functionality will be provided, in order preventing producing matrixes without explicit description of what it contains. It indicates
  - the kind of elements provided as source (lines) and target (columns), plus the considered type of relationships (all the visible ones per default, but it is possible to constraints the ones which are displayed)
  - Value for cell colors indicate what was used in order to define the colors, by default the Number of relations, which is the only available for this version. Several others will be proposed in the next versions, such as value which allows to calculate the shortest path or any other relevant typed property for coloring the cells.
  - display of roots or nodes

When clicking a cell, the number and the list of relationships is displayed.

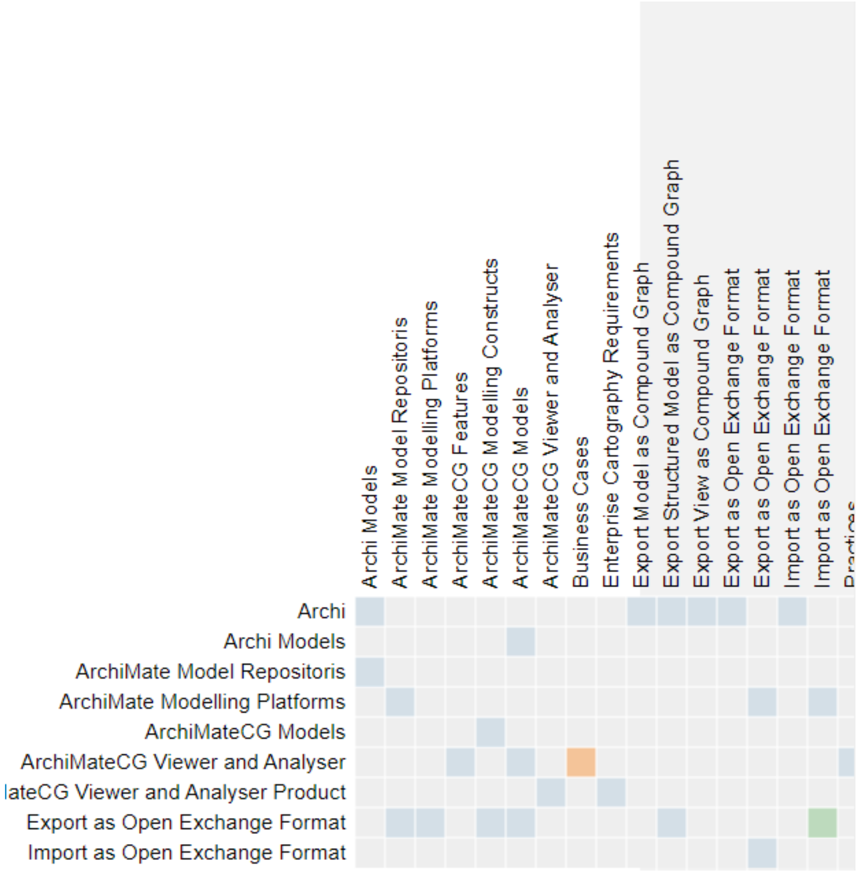

Archi Models
ArchiMate Model Repositoris
ArchiMate Modelling Platforms
ArchiMateCG Features
ArchiMateCG Modelling Constructs
ArchiMateCG Models
ArchiMateCG Viewer and Analyser
Business Cases
Enterprise Cartography Requirements
Export Model as Compound Graph
Export Structured Model as Compound Graph
Export View as Compound Graph
Export as Open Exchange Format
Export as Open Exchange Format
Import as Open Exchange Format
Import as Open Exchange Format
Practices
Archi
Archi Models
ArchiMate Model Repositoris
ArchiMate Modelling Platforms
ArchiMateCG Models
ArchiMateCG Viewer and Analyser
lateCG Viewer and Analyser Product
Export as Open Exchange Format
Import as Open Exchange Format

Figure 13: Current representation of the graph as matrix

Limitations

Currently, all the visible edges are considered, including those which are 1) collapsed edges, which are representing a set of edges between a source and a target in a more "condensed" way. 2) Edges between upper visible collapsed parents of contained elements. This makes the produced matrixes not always accurate. So when clicking a node, the list doesn't consider the collapsed edges or edges created between upper visible collapsed parents of related nodes. This opens the question of distinguishing the actual relations and the edges which are not actual relations, created for simplified visualisations of the compound graphs representing the model.

When some nodes don't have labels, no label is provided as line or column header. It should be managed for more visibility.

There is no way to distinguish in the production of the matrix between compound nodes and simple nodes on the matrix. There is also no way to distinguish between containments representing nested model elements (e.g. a model element within a package) or a composition relationship. This should be considered when building the matrix.

So at this stage, the matrix tool is mainly useable when having no collapsed edges and no edges derived from relationships between nested nodes within collapsed parents.

For responding to precise questions of an architect, it should be considered some types of model elements as sources and some types of model elements as target, and eventually some type(s) of relations or nesting. It is not possible to parameterize them from the matrix tool, the filtering is to be made before on the graph, showing only what we want to be considered.

Planned extensions

- *Considerations of edges representing relationships and those resulting from graphs with some collapsing (cf. limitations)*
- *Ability to filter and address separately the edges resulting from collapsing*
- *Extended Dynamic filtering of the matrix content*
- *More choices for Rows/columns sorting*
- *Matrix for compound graphs with collapse and expand capabilities, similar to the graph one.*
- *More properties considered for values calculation or display at the intersection of the lines and columns.*
- *Parameterizing from the toolbar or from the palette which types of elements are in the matrix*
- *Integration of legends, which should be Property Colouring Tool similar…*

Colouring

This tool provides a very simple feature: ability to choose a colour to be applied to the background of the selected nodes.

It is planned to extend it in order to provide more features for changing the colours of graph elements, driven by the value it brings.

In the following figure, you can see the button for choosing the colour to be applied to a selection. Or this, click on the button for "Color for selected", and it will make a colour chooser appear. Then select elements of the graph and click on the Tag button. It will change the background colour of expanded compound nodes.

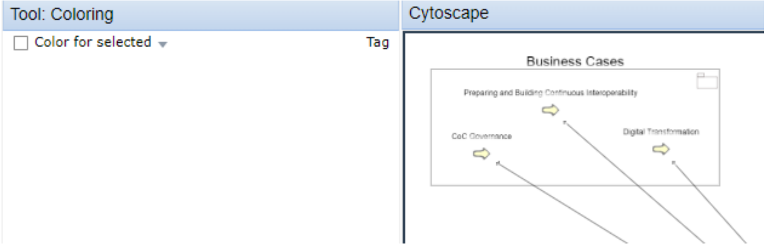


Figure 14: Colouring toolbar

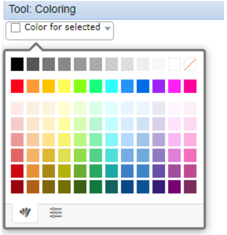


Figure 15: Colour selector

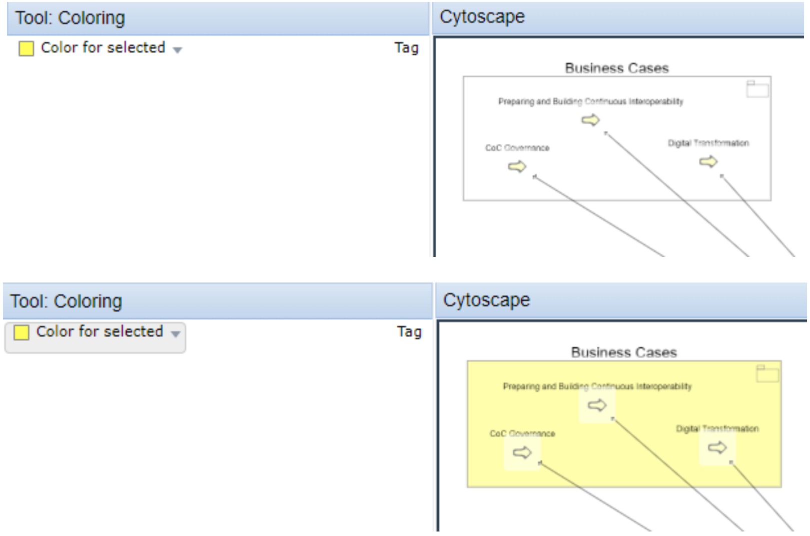
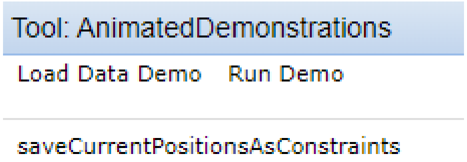


Figure 16: Before and after applying the colour on selection

Animated Demonstration

The libraries used for developing ArchiMateCG come with some animation functionalities. They were used in order to provide some dynamic demonstration of the usage of ArchiMate. For this, an initial graph is to be defined, on top of which different actions will be registered in order to make it possible to replay it automatically. It can be used for communication purposes, for training and eventually in the future for simulation.

When selecting the "Animated Demonstration" tool, the proposed toolbar is the one on the image at the left side. First Data for Demo are to be loaded. For this version, a single data set is available (related to the Digital Design, Manufacturing and Support Airbus Initiative, and in particular to the Reference built relying on ArchiMate)


Tool: AnimatedDemonstrations
Load Data Demo
Run Demo
saveCurrentPositionsAsConstraints

When the data is loaded, it is then possible to launch the animation by clicking on the "Run Demo" button. Unlike movies, animation consists in acting on actual data, applying different kinds of animations/transformations on the graph visual aspects (zooming, moving, positioning, etc.) in combination with a timer.

The fully collapsed graph currently available is represented in Figure 17

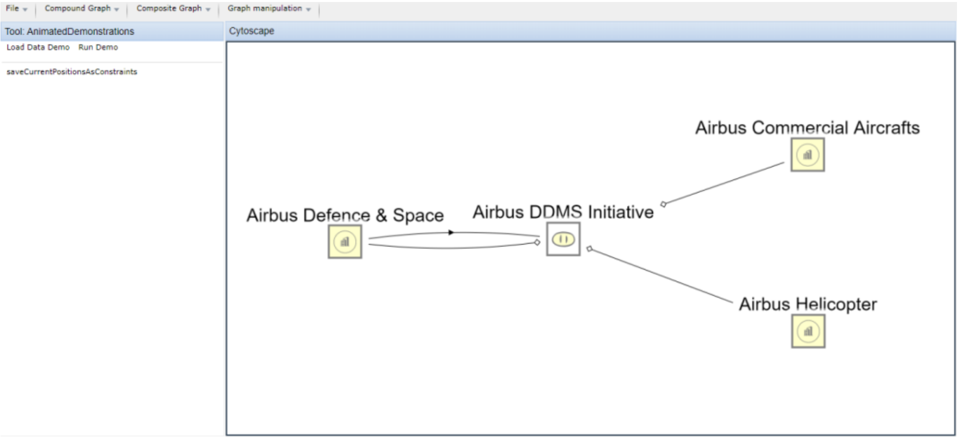


Figure 17: Animated Demonstration toolbar

When fully expanded, it looks like in Figure 18

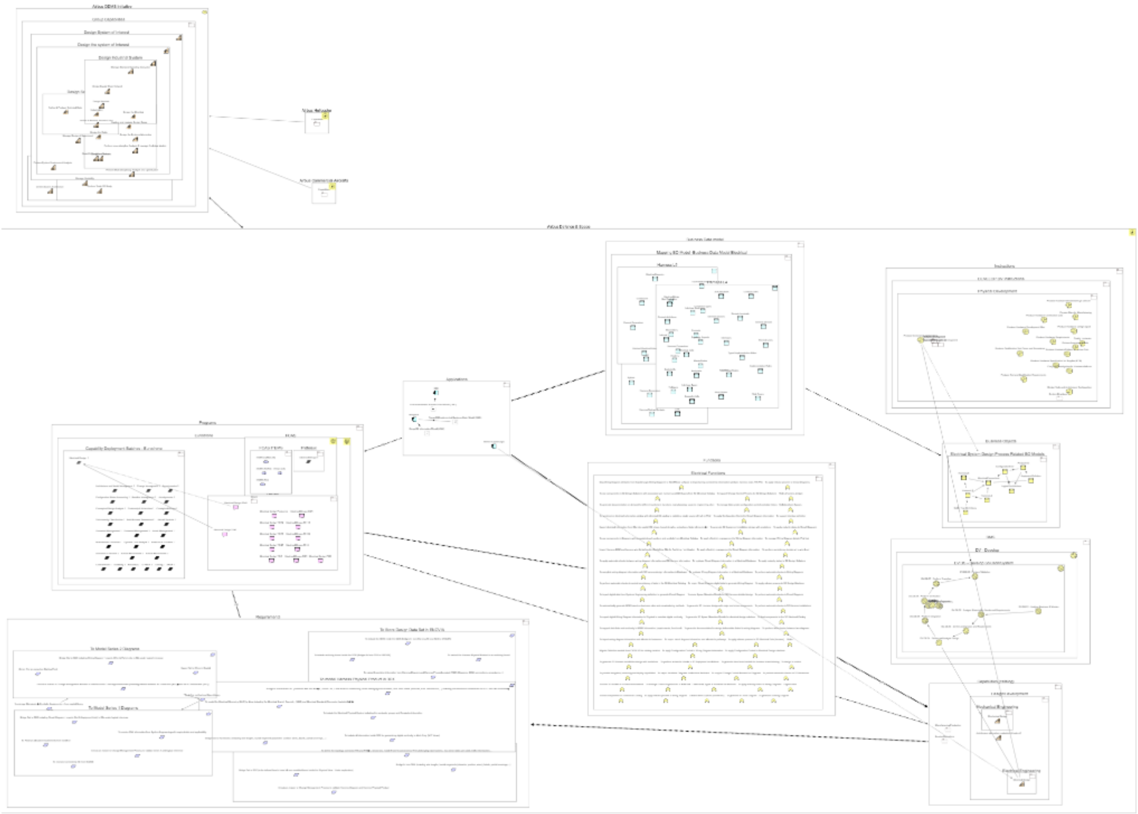


Figure 18: Expanded demonstration model

The proposed animation shows navigation in the graph, with successive expansions, collapsing and zooming, associated with some story telling about objectives and added value of the Reference created for DDMS for programs. Here capabilities for Electrical Design are explored, with the links to business applicable processes, procedures and instructions, being one division specific or for the whole Group. As applied to living data, it gives an actual flavour concerning the performed work, but also about what compound graphs are and how they can be used for analytics, relying on semantic cartography.

It is planned in the future to provide for with this tool a list of demonstrations which will be selectable, and with each of them a starting graph and the animation sequence ready to be played.

It is also planned to provide functionalities for supporting creation of such animation for non-programmers, e.g. providing functionalities such as the one currently available for saving current position of graph nodes as constraints to be used in the animations.

savingCurrentPositionAsConstraints

It saves a JSON file with positions as constraints, which can be used with FCOSE layout used on top of ArchiMateCG. It allows you to master in details the expected positioning at the end of some animation sequences.

The work related to animation remains exploratory, for value and feasibility assessment.

Fcose

Fcose is the main ArchiMateCG used layout among those proposed for Cytoscape.js.

“fCoSE (pron. "f-cosay", fast Compound Spring Embedder), is a faster version of our earlier compound spring embedder algorithm named CoSE, implemented as a Cytoscape.js extension by i-Vis Lab in Bilkent University… fCoSE layout algorithm combines the speed of spectral layout with the aesthetics of force-directed layout. fCoSE runs up to 2 times as fast as CoSE while achieving similar aesthetics.” (fCose GitHub project)

This Cytoscape.js layout extension comes with many options, which are not always easy to understand when willing to parameter it properly for usage with ArchiMateCG, despite some online demonstrators. Default options are:

- **Quality**: "default"- Use random node positions at beginning of layout if this is set to false, then quality option must be "proof". Possible values are 1)'draft' which only applies spectral layout 2)'default' which improves the quality with incremental layout (fast cooling rate) 3) 'proof' which improves the quality with incremental layout (slow cooling rate)
- **Randomize**: true - Whether or not to animate the layout
- **Animate**: true - Duration of animation in ms, if enabled animationDuration: 1000
- **Easing** of animation, if enabled **animationEasing**: undefined
- **Fit**: true - Fit the viewport to the repositioned nodes
- **Padding**: 30- Padding around layout
- **NodeDimensionsIncludeLabels**: false - Whether to include labels in node dimensions. Valid in "proof" quality
- **UniformNodeDimensions**: false - Whether or not simple nodes (non-compound nodes) are of uniform dimensions
- **PackComponents**: true - Whether to pack disconnected components - Cytoscape-layout-utilities extension should be registered and initialized
- **step**: "all"- Layout step - all, transformed, enforced, cose - for debug purpose only

/* spectral layout options */

- **samplingType**: true -False for random, true for greedy sampling
- **sampleSize**: 25 - Sample size to construct distance matrix
- **nodeSeparation**: 75 - Separation amount between nodes
- **piTol**: 0.0000001- Power iteration tolerance

/* incremental layout options */

- **nodeRepulsion**: node => 4500 - Node repulsion (non overlapping) multiplier
- **idealEdgeLength**: edge => 50 - Ideal edge (non nested) length
- **edgeElasticity**: edge => 0.45 - Divisor to compute edge forces
- **nestingFactor**: 0.1 - Nesting factor (multiplier) to compute ideal edge length for nested edges
- **numIter**: 2500 -Maximum number of iterations to perform - this is a suggested value and might be adjusted by the algorithm as required
- **tile**: true – for enabling tiling
- **tilingPaddingVertical**: 10 - Represents the amount of the vertical space to put between the zero degree members during the tiling operation(can also be a function)
- **tilingPaddingHorizontal**: 10 - Represents the amount of the horizontal space to put between the zero degree members during the tiling operation(can also be a function)
- **gravity**: 0.25 - Gravity force (constant)
- **gravityRangeCompound**: 1.5 - // Gravity range (constant) for compounds
- **gravityCompound**: 1.0 - Gravity force (constant) for compounds
- **gravityRange**: 3.8 -Gravity range (constant)
- **initialEnergyOnIncremental**: 0.3 - Initial cooling factor for incremental layout

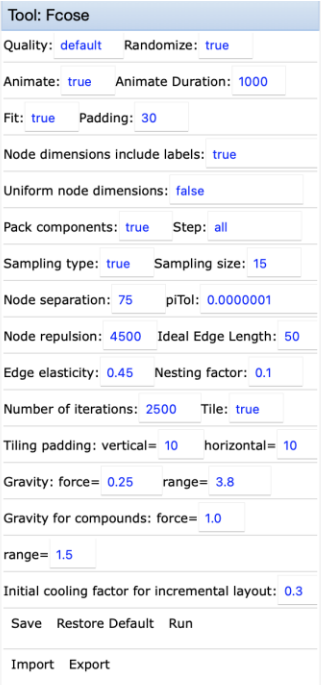
/* constraint options */

- fixedNodeConstraint: undefined - Fix desired nodes to predefined positions
  [{nodeId: 'n1', position: {x: 100, y: 200}}, {...}]
- alignmentConstraint: undefined - Align desired nodes in vertical/horizontal direction
  {vertical: [['n1', 'n2'], [...]], horizontal: [['n2', 'n4'], [...]]}
- relativePlacementConstraint: undefined - // Place two nodes relatively in vertical/horizontal direction
  [{top: 'n1', bottom: 'n2', gap: 100}, {left: 'n3', right: 'n4', gap: 75}, {...}]

Note: fixed node constraints are widely used for producing ArchiMateCG animations.

The goal of this tool is to be able to modify and to apply Fcose parameters, in order to experiment the effect. It is targeted to be able to save and to load the defined options in order the users to parameter Fcose layout as they wish, without having to program. It will also be possible sharing the parameterization files between users.

Note: as this time, “Import” and “Export” are not yet implemented, it will in a future version.



Tool: Fcose

Quality: default Randomize: true

Animate: true Animate Duration: 1000

Fit: true Padding: 30

Node dimensions include labels: true

Uniform node dimensions: false

Pack components: true Step: all

Sampling type: true Sampling size: 15

Node separation: 75 piTol: 0.0000001

Node repulsion: 4500 Ideal Edge Length: 50

Edge elasticity: 0.45 Nesting factor: 0.1

Number of iterations: 2500 Tile: true

Tiling padding: vertical= 10 horizontal= 10

Gravity: force= 0.25 range= 3.8

Gravity for compounds: force= 1.0

range= 1.5

Initial cooling factor for incremental layout: 0.3

All the parameters are those provided in the options which can be defined for Fcose. The initial value are the default values for these parameters.

When modified, it is possible them to “save” them: they will replace the current parameters for Fcose for the current running ArchiMateCG session (i.e. till it will be closed or reloaded with the Web navigator).

The initial default values can be restored using the button “Restore default”.

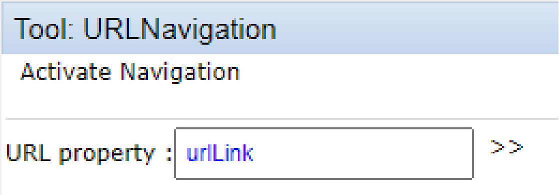
“Run” will apply the layout on the current graph. It allows to see the differences coming from the change made on the parameters.

Those parameters are documented on the Fcose GitHub project, but is not so easy to clearly understand what their effect are. Making it possible to experiment before to keep the proper configuration is the intent of the Fcose tool.

URL Navigation tool

This tools aims at activating URL navigation and at indicating which property of nodes or edges is to be used for navigation.

By default, the navigation is not activated. This is due in particular to the potential security constraints making navigation to external resources forbidden. It can be activated (and deactivated) using the top menu Parameters>>URL Navigation.



Tool: URLNavigation

URL property : urlLink

The button “Activate Navigation” is an alternative way to activate the navigation.

The field “URL property” allows to change the name of the property used in order to find the URL which will be open by double clicking. Let’s click on >> in order to register the change.

Timeline tool

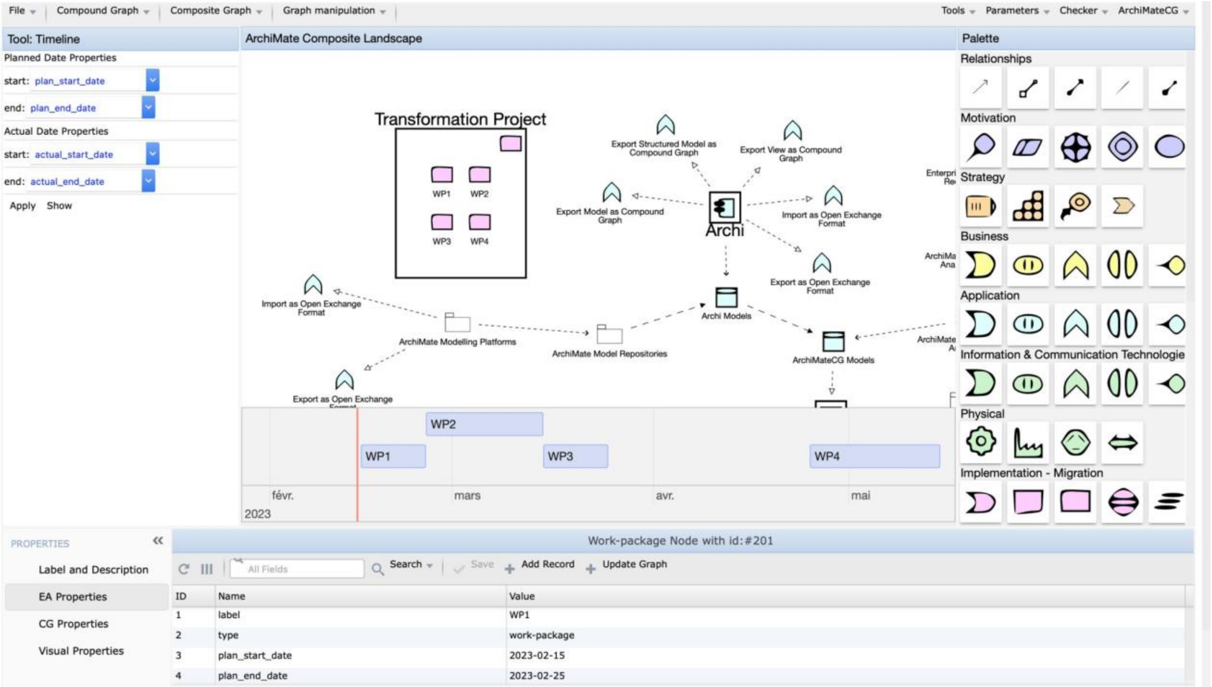
This is a tool proposed since version 1.0.20. It aims at exploring time aspect with Enterprise Architecture and ArchiMateCG.

This first alpha version proposes a pane with a selector for each time related properties: planned start and end data, and actual start and end data, with a proposed default and only proposed names “plan_start_date”, “plan_end_date”, “actual_start_date” and “actual_end_date”.

When pushing “Apply” button, it creates the timeline with as content all the model elements which have these two values fulfilled. The format must be aaaa-mm-dd (aaaa for year, mm for month and dd for day). In order to display or to hide it, the “Show” button is to be used.

The timeline is the integration of the timeline of vis.js. When having the mouse over the timeline and rolling down or up, it changes the time scale. When clicking on the timeline having the click button down, it is possible to make the timescale moving back in the past of forward in the future.

The size of the timeline feature is 25% of the height. When many elements, it is possible to make the timescale move up or down when overflow.


File
Compound Graph
Composite Graph
Graph manipulation
Tools
Parameters
Checker
ArchiMateCG
ArchiMate Composite Landscape
Transformation Project
WP1
WP2
WP3
WP4
Export Structured Model as Compound Graph
Export View as Compound Graph
Export Model as Compound Graph
Archi
Import as Open Exchange Format
Export as Open Exchange Format
Archi Models
Import as Open Exchange Format
ArchiMate Modelling Platforms
ArchiMate Model Repositories
ArchiMateCG Models
Export as Open Exchange Format
févr.
mars
avr.
mai
2023
Palette
Relationships
Motivation
Strategy
Business
Application
Information & Communication Technologie
Physical
Implementation - Migration
PROPERTIES
Label and Description
EA Properties
CG Properties
Visual Properties
All Fields
Search
Save
Add Record
Update Graph

Tool: Timeline

Planned Date Properties

start: plan_start_date

end: plan_end_date

Actual Date Properties

start: actual_start_date

end: actual_end_date

Apply Show

Work-package Node with id:#201

ID	Name	Value
1	label	WP1
2	type	work-package
3	plan_start_date	2023-02-15
4	plan_end_date	2023-02-25

Figure 19: Timeline Tool

New features will be added driven by value creation, relying on the visualisation component integrated so far and being a basis for capturing needs.

Usage for architects dealing with Architectural representation in ArchiMate

ArchiMateCG provide a way to represent an ArchiMate model as a compound interactive graph, with as underlying metamodel respectively nodes and edges which are respectively typed with ArchiMate element types and ArchiMate relationship types. From a visual point of view, the rendering of nodes is made by using an icon with the symbols defined for model elements by ArchiMate (Figure 20). Concerning the edges, they are rendered as arcs with the same visual representation which is specified for the different kinds of relationships.

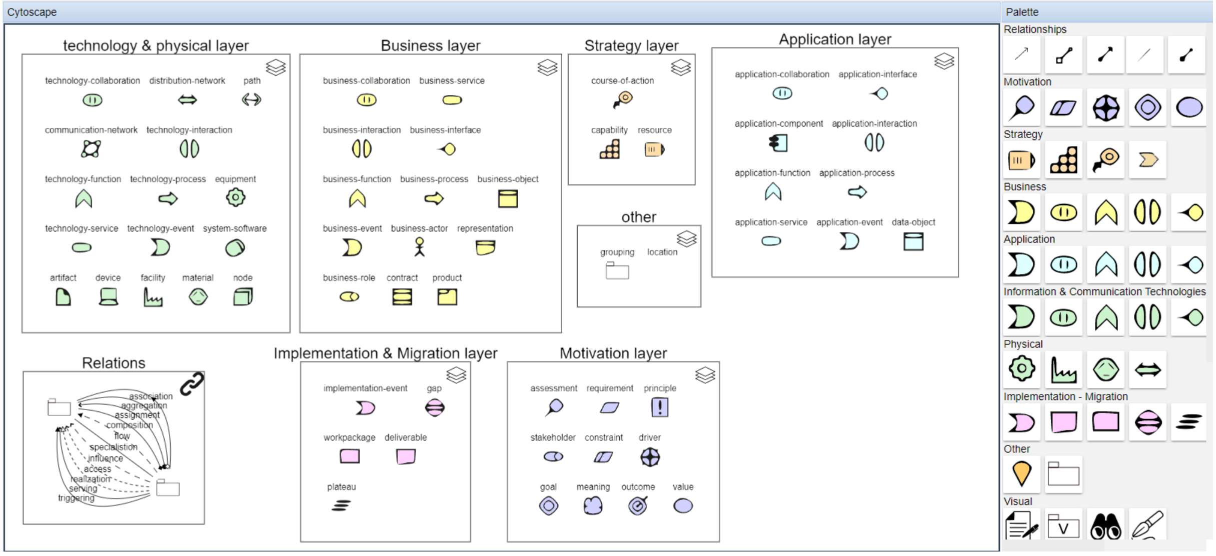


Figure 20: All the ArchiMate constructs as represented with ArchiMateCG

All ArchiMate constructs can be represented, except relationships between a model element and a relationship. The constraints related to ArchiMate relationships are not enforced by the ArchiMateCG for this version.

The model provided when opening ArchiMateCG provided an architectural representation of ArchiMateCG in ArchiMate, using ArchiMate visual symbols for model elements (nodes) and relationships (edges).

Nodes with a black square are compound nodes: they can be expanded in order to show model elements they contain, or reflecting composition relationship, or reflecting containment only (mainly for physical model containments)

E.g. Figure 21, Figure 22and Figure 23 shows the “ArchiMateCG Features” compound node with all the menus, represented as ArchiMate Grouping, which will contain one ArchiMate application function per menu item.

So visual mining is possible combining nodes and edges collapsing and expanding, filtering, locking, grabbing and removing. It is also possible to add new model elements to the cartography, being nodes or edges, collapsed or expanded.

Let’s note that all the icons are SVG icons, and that they are hardcoded. It is not possible at this stage to reference icons with a URI, due to the adopted design principles driven by security constraints.

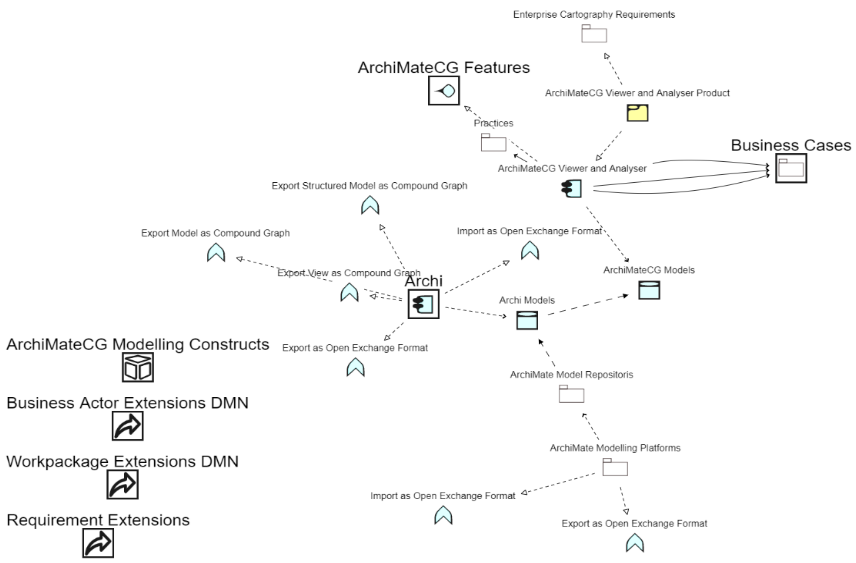

Enterprise Cartography Requirements
ArchiMateCG Features
ArchiMateCG Viewer and Analyser Product
Practices
Business Cases
ArchiMateCG Viewer and Analyser
Export Structured Model as Compound Graph
Export Model as Compound Graph
Import as Open Exchange Format
ArchiMateCG Models
Export View as Compound Graph
Archi
Archi Models
ArchiMateCG Modelling Constructs
Export as Open Exchange Format
Business Actor Extensions DMN
ArchiMate Model Repositoris
Workpackage Extensions DMN
ArchiMate Modelling Platforms
Import as Open Exchange Format
Requirement Extensions
Export as Open Exchange Format

Figure 21: Fully collapse ArchiMateCG initial model

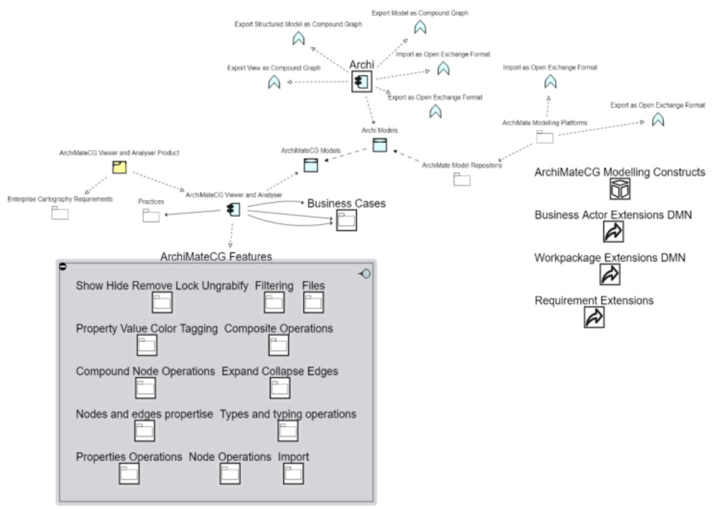

Export Structured Model as Compound Graph
Export Model as Compound Graph
Import as Open Exchange Format
Export View as Compound Graph
Archi
Import as Open Exchange Format
Export as Open Exchange Format
Export as Open Exchange Format
Archi Models
ArchiMate Modelling Platforms
ArchiMateCG Viewer and Analyser Product
ArchiMateCG Models
ArchiMate Model Repositoris
ArchiMateCG Modelling Constructs
Enterprise Cartography Requirements
Practices
ArchiMateCG Viewer and Analyser
Business Cases
Business Actor Extensions DMN
ArchiMateCG Features
Workpackage Extensions DMN
Show Hide Remove Lock Ungrabify
Filtering
Files
Requirement Extensions
Property Value Color Tagging
Composite Operations
Compound Node Operations
Expand Collapse Edges
Nodes and edges propertise
Types and typing operations
Properties Operations
Node Operations
Import

Figure 22: the same graph after expanding the "ArchiMateCG Features" compound node

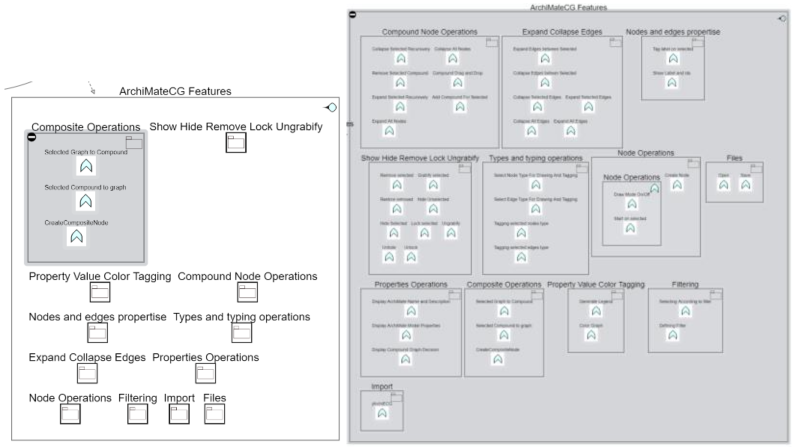


Figure 23: the same graph after expanding all the "ArchiMateCG Features" sub compound nodes

ArchiMateCG also provides some visual modelling constructs which are not part of the visual language. It provides meta-concepts and group of elements, such as viewpoints, not defined types, extensions supported by ArchiMateCG, Relation constructs, ArchiMateCG Metamodel, layer, models, folders or packages. They are represented in the same order in Figure 24.

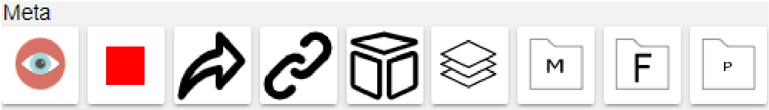


Figure 24: Meta concept symbols

Some other visual elements available are proposed, as Archi notes, ArchiMate groups, ArchiMate views and Archi drawings. They are represented in the same order in

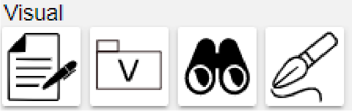


Figure 25: Visual elements symbols

Finally some extensions are proposed for ArchiMate Work Package (Program and Project), ArchiMate Business Processes (Process, Procedure and Instruction), ArchiMate Business Actors (Digital Business Ecosystem, Networked Organisation, Enterprise, Department, (Organisational) Service, Team and Person) and ArchiMate Requirements (Internal Requirement, External Requirement, Functional Requirement and Non Functional Requirement).

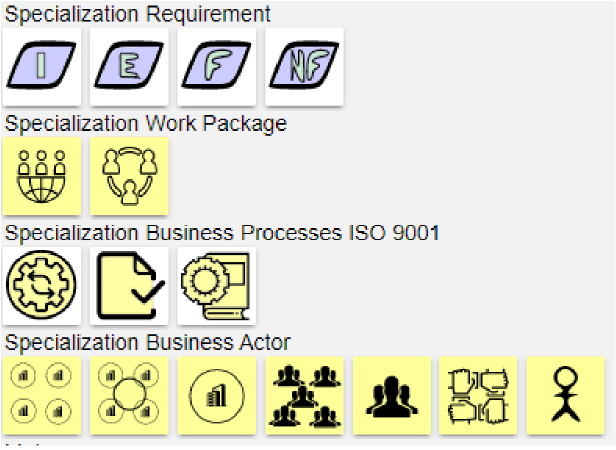

Specialization Requirement
I
E
F
NF
Specialization Work Package
Specialization Business Processes ISO 9001
Specialization Business Actor

Figure 26: illustration of ArchiMate language extensions

So the produced models can also provide complementary information to architects, concerning the physical breakdown structure of models and the logical organisations of views by means of viewpoints.

The underlying implementation metamodel (Figure 27) is an extension of the metamodel of Cytoscape.js, which is the software component used for Compound Graph Visualisation and Graph Processing:

All nodes and edges come with a “Type” property, which is given an ArchiMate type or ArchiMateCG complementary type.

Compound nodes are extended for Composite nodes, with a component node being the parent of a compos ite node, composite and component typing depending on an existing composition relationship represented by an edge, and reference with the object property “parentRelationId”.

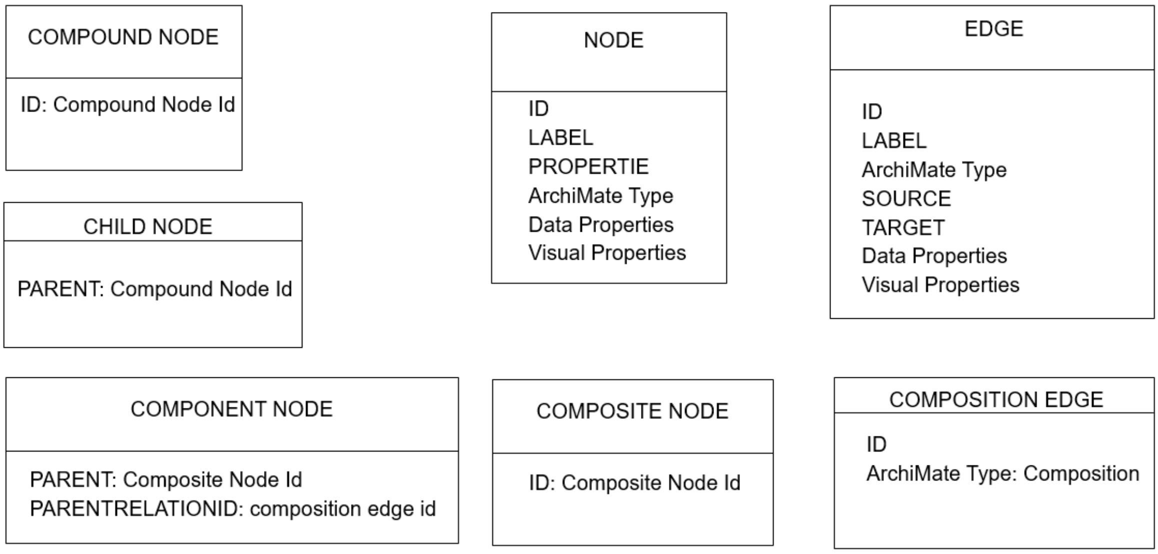


Figure 27: simple ArchiMateCG implementation metamodel

The serialisation formats used for data import/export and file saving are all based on this information model, which also indicate how ArchiMateCG extended Cytoscape.js.

ArchiMateCG has been iteratively developed in an agile way through several iterations with continuous extension and assessment of Interoperability and value creation.

Usage based on graph and network theories

A few functionalities have been provided so far taking advantage of those provided by Cytoscape.js.

Filtering with filter expression

A field is provided to provide an expression from which some nodes and edges will be selected. Several filters can be applied successively, without deselecting the result of the previous research.

The filtering expressions are the same as those defined by Cytoscape.js.

It can concern nodes or edges, and some condition on data properties. "node" or "edge" are to be indicated if the nature of the graph element is a filter. * is used for both nodes and edges.

Properties are indicated encapsulated by [].

The property name is given, with the comparison expression (=; >, <, >=, <=, and the considered value between quotes.

E.g. `node[name = "Jerry"]`

#id matches elements with the same matching Id.

Concerning data properties, the following can be used:

- [name] Matches elements if they have the specified data attribute defined, i.e. not undefined (e.g. [foo] for an attribute named "foo"). Here, null is considered a defined value.
- [^name] Matches elements if the specified data attribute is not defined, i.e. undefined (e.g [^foo]). Here, null is considered a defined value.
- [?name] Matches elements if the specified data attribute is a truthy value (e.g. [?foo]).
- [!name] Matches elements if the specified data attribute is a falsy value (e.g. [!foo]).
- [name = value] Matches elements if their data attribute matches a specified value (e.g. [foo = 'bar'] or [num = 2]).
- [name != value] Matches elements if their data attribute doesn't match a specified value (e.g. [foo != 'bar'] or [num != 2]).
- [name > value] Matches elements if their data attribute is greater than a specified value (e.g. [foo > 'bar'] or [num > 2]).
- [name >= value] Matches elements if their data attribute is greater than or equal to a specified value (e.g. [foo >= 'bar'] or [num >= 2]).
- [name < value] Matches elements if their data attribute is less than a specified value (e.g. [foo < 'bar'] or [num < 2]).
- [name <= value] Matches elements if their data attribute is less than or equal to a specified value (e.g. [foo <= 'bar'] or [num <= 2]).
- [name *= value] Matches elements if their data attribute contains the specified value as a substring (e.g. [foo *= 'bar']).
- [name ^= value] Matches elements if their data attribute starts with the specified value (e.g. [foo ^= 'bar']).
- [name $= value] Matches elements if their data attribute ends with the specified value (e.g. [foo $= 'bar']).
- [name.0 = value] Matches elements if their data attribute is an array and the element at the defined index matches a specified value (e.g. [foo.0 = 'bar']).
- [name.property = value] Matches elements if their data attribute is an object and the property with the defined name matches a specified value (e.g. [foo.bar = 'baz']).

- @ (data attribute operator modifier) Prepended to an operator so that is case insensitive (e.g. [foo @$= 'ar'], [foo @>= 'a'], [foo @= 'bar'])
- ! (data attribute operator modifier) Prepended to an operator so that it is negated (e.g. [foo !$= 'ar'], [foo !>= 'a'])

Concerning compound nodes:

- “>”(child selector) Matches direct children of the parent node (e.g. node > node).
- $ (subject selector) Sets the subject of the selector (e.g. $node > node to select the parent nodes instead of the children).
- :parent : Matches parent nodes (they have one or more child nodes).
- :childless : Matches childless nodes (they have zero child nodes).
- :child or :nonorphan: Matches child nodes (they each have a parent).
- :orphan : Matches orphan nodes (they each have no parent).
- :compound : Matches parent nodes. Also matches edges connected to parent nodes (they each have at least one parent node among source and target).

Edges:

- :loop : Matches loop edges (same source as target).
- :simple : Matches simple edges (i.e. as would be in a simple graph, different source as target).

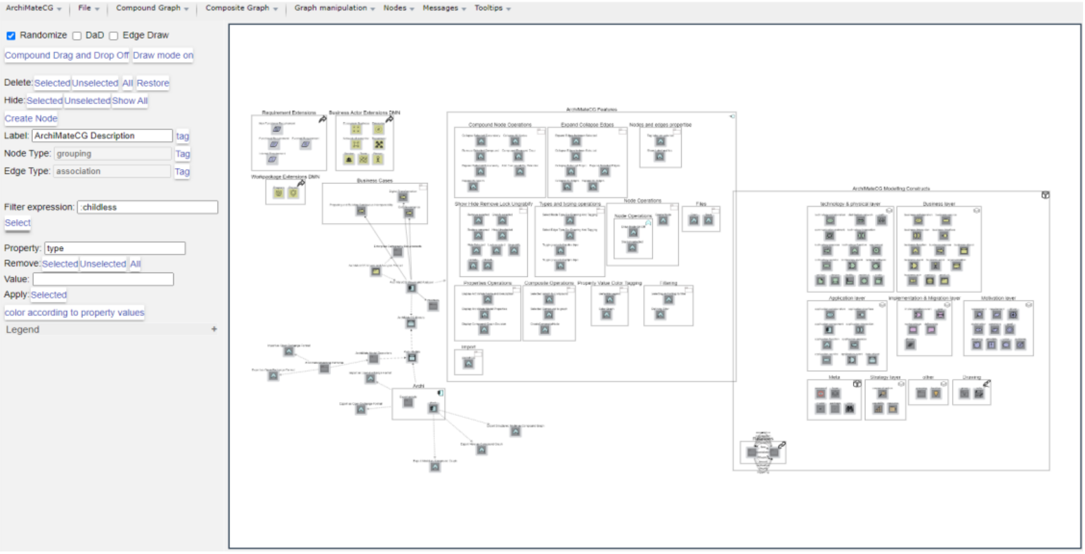
Note 1: when there is an error in the expression, all the visible nodes are selected. Deselect by clicking on the graph zone with node graph element.

Note 2: successive filtering are cumulative in terms of selection

Note 3: selection by filtering can be combined with graph operations for removing, locking, hiding, grouping with a compound node or tagging.

Some examples:

- Childless nodes


ArchiMateCG
File
Compound Graph
Composite Graph
Graph manipulation
Nodes
Messages
Tooltips
Compound Drag and Drop Off
Draw mode on
Delete: Selected Unselected All Restore
Hide: Selected Unselected Show All
Create Node
tag
Tag
Tag
Select
Remove: Selected Unselected All
Apply: Selected
color according to property values
Legend
Requirement Extensions
Business Actor Extensions DMN
Workpackage Extensions DMN
Business Cases
ArchiMateCG Features
Compound Node Operations
Show Hide Remove Lock Ungrabify
Properties Operations
Composite Operations
Import
Archi
ArchiMateCG Modelling Constructs

☑ Randomize ☐ DaD ☐ Edge Draw

Label: ArchiMateCG Description

Node Type: grouping

Edge Type: association

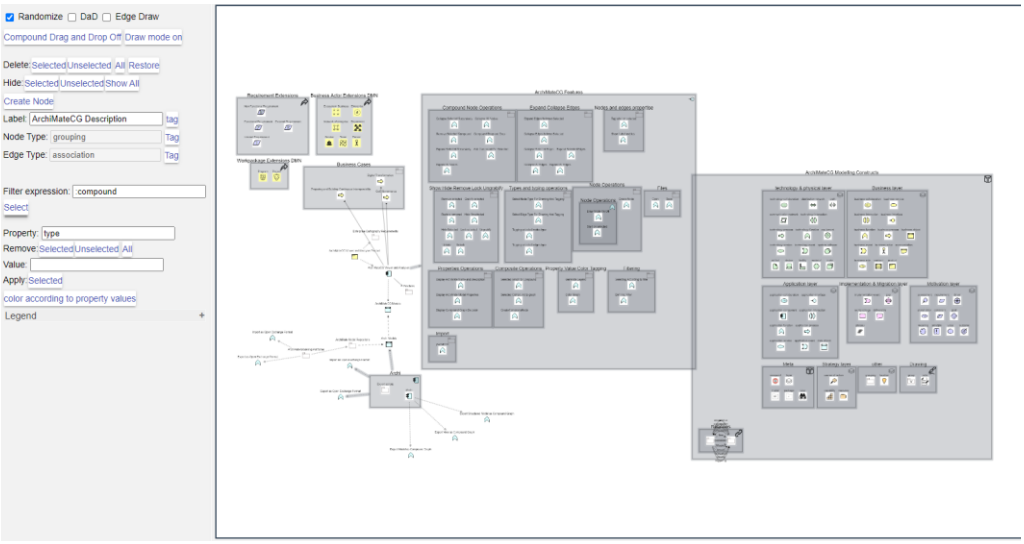
Filter expression: :childless

Property: type

Value:

Figure 28: Selection of childless nodes with the selector ad filter expression

- Compound



☑ Randomize ☐ DaD ☐ Edge Draw

Compound Drag and Drop Off Draw mode on

Delete: Selected Unselected All Restore

Hide: Selected Unselected Show All

Create Node

Label: ArchiMateCG Description tag

Node Type: grouping Tag

Edge Type: association Tag

Filter expression: :compound

Select

Property: type

Remove: Selected Unselected All

Value:

Apply: Selected

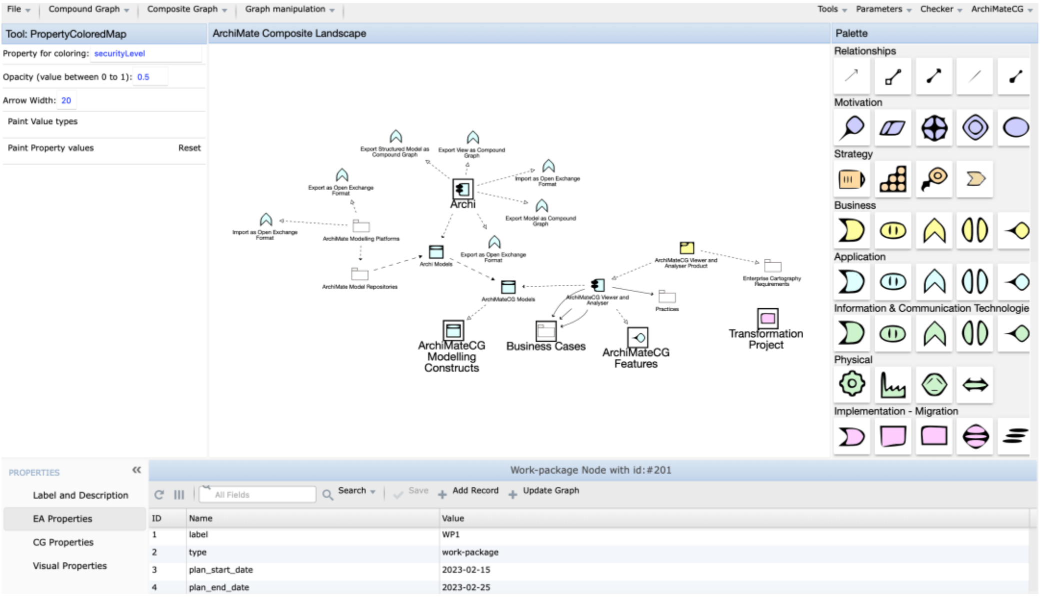
color according to property values

Legend +

Figure 29: Selection of compound nodes with the selector ad filter expression

Colourizing a graph according to the values of a given data property

After selecting Tools>ColoredMap, a new toolbar is proposed on the left pan. It allows to enter the name of a node property, and then, to click on.



Tool: PropertyColoredMap

Property for coloring: securityLevel

Opacity (value between 0 to 1): 0.5

Arrow Width: 20

Paint Value types

Paint Property values Reset

Work-package Node with id:#201

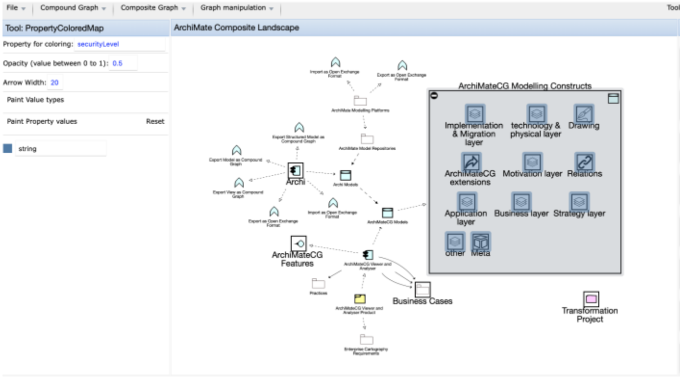
ID	Name	Value
1	label	WP1
2	type	work-package
3	plan_start_date	2023-02-15
4	plan_end_date	2023-02-25

Figure 30: Property Coloured Map Tool panel

1. The button "Paint Value types"

As a result

- A legend is automatically created with an automated assignment of a colour for each JSON data type of the value in the list (Boolean, integer, string or float). This allows to ensure that the property value is homogeneously typed. It may occur when importing external data sources where it was not enforced. Using this button allows to check it.
- All the nodes and arcs will be coloured according to the value type for the property, and left blank if the property is not provided to the model element.



Tool: PropertyColoredMap

Property for coloring: securityLevel

Opacity (value between 0 to 1): 0.5

Arrow Width: 20

Paint Value types

Paint Property values Reset

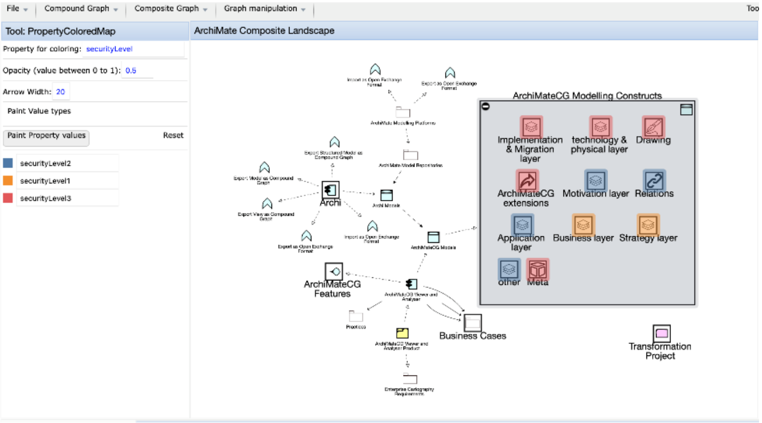
🟦 string

Figure 31: Coloured map according to the types of the property

2. The button "Paint Property values".

As a result:

- A legend is automatically created with an automated assignment of a colour for each value of the property
- All the nodes and arcs will be coloured according to their value for the property, and left blank if the property is not provided to the model element


File
Compound Graph
Composite Graph
Graph manipulation
Too
ArchiMate Composite Landscape
Import as Open Exchange Format
Export as Open Exchange Format
ArchiMate Modelling Platforms
Export Structured Model as Compound Graph
ArchiMate Model Repositories
Export Model as Compound Graph
Archi
Archi Models
Export View as Compound Graph
Import as Open Exchange Format
Export as Open Exchange Format
ArchiMateCG Models
ArchiMateCG Features
ArchiMateCG Viewer and Analyser
Practices
ArchiMateCG Viewer and Analyser Product
Business Cases
Enterprise Cartography Requirements
ArchiMateCG Modelling Constructs
Implementation & Migration layer
technology & physical layer
Drawing
ArchiMateCG extensions
Motivation layer
Relations
Application layer
Business layer
Strategy layer
other
Meta
Transformation Project

Tool: PropertyColoredMap

Property for coloring: securityLevel

Opacity (value between 0 to 1): 0.5

Arrow Width: 20

Paint Value types

Paint Property values   Reset

🟦	securityLevel2
🟧	securityLevel1
🟥	securityLevel3

Figure 32: Coloured map according to the values of the selected property

The goal is to perform some analysis of a compound model graph in a visual and easy way (visual mining).

Two parameters can be provided:

- Opacity (value between 0 and 1): it allows to change the opacity of the applied colours for the nodes
- Arrow Width: it allows to change the width of the arrows in order to visualize better their colour.

These parameters can be changed and applied by clicking again on the button “Colors for property value”.

It is possible to reset (button “Reset”) the graph, which remove the background colours for nodes, put the edge colour to black and the edge width to 1.

So it is possible to visually explore a graph according some properties of the nodes and of the edges. In the initial dataset coming with ArchiMateCG, the property securityLevel is added to some elements of the graph. The provided figure illustrates what happens when selecting this property for coloured map: a legend is created with a dedicated colour per existing value (automatically), and the nodes and edges are colours, taking into consideration the edge width and opacity parameters.

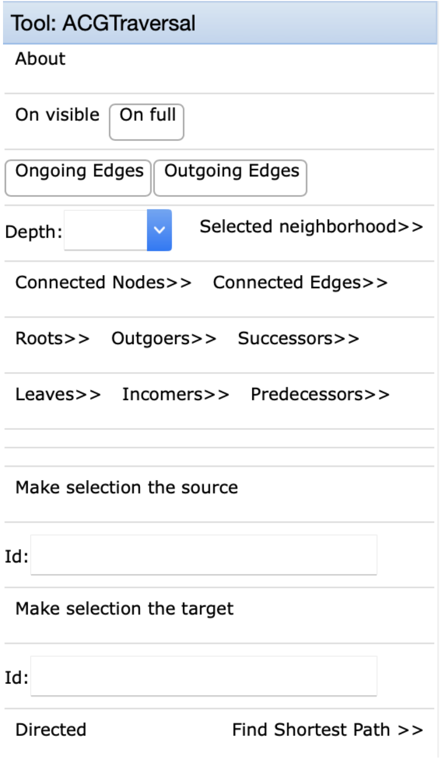
Note 1: since version 1.0.20, the graph is not reorganized anymore when colorizing the graph.

Note 2: the legend is mainly relevant for text value with a limited number of values, or for numeric values with a limited number of values, without continuity for the considered property. It is planned in future version to support scrolling when many value to be displayed in the legend. It is also planned to propose Colour(s) gradients appropriate legend for properties with continuous value (e.g. age when we don’t want one colour per age given in the properties but a gradient reflecting the age is little or great).

Searching with graph analytics

This is made available through the tool ACGTraversal you can select on the Tools menu.

When selected, the following toolbar appears:



Clicking on “About” shows information about the tool.

The two next buttons allow to indicate if proposed operations will be done on the visible graph or on the full graph, i.e. considering hidden graph elements.

The two next buttons allow to parameter if we are considering Ongoing edges, Outgoing edges or both for the various operations performs with this tool.

Then the Neighbourhood selection operation is propose. This is perform on the current selection on the graph. It is needed to indicate the depth (1 to 9). Let’s note that it’s also possible to click several times on the “Selected neighbourhood” button for the same effect.

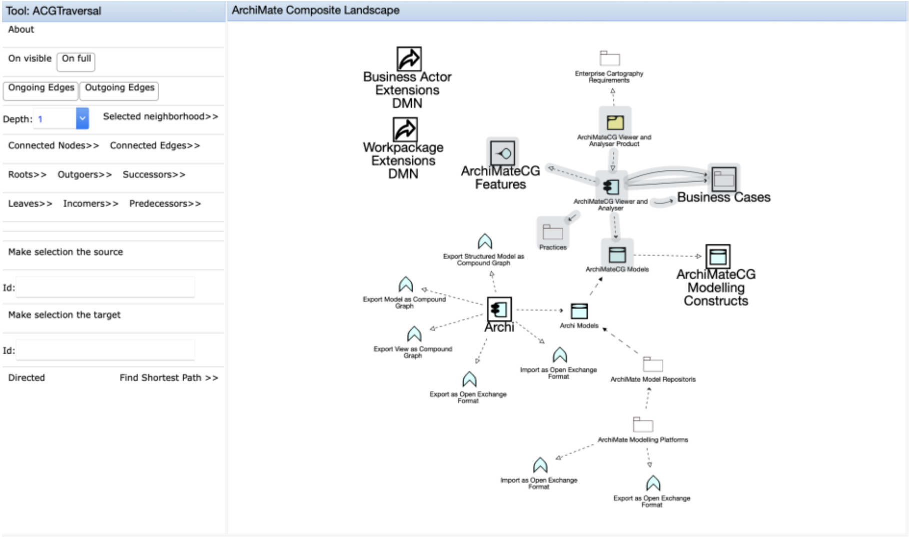
Then you have a set of buttons allowing to find on the selected nodes those which are connected nodes, connected edges, roots, out goers, successors, leaves, incomers or predecessors. For roots and leaves, all the nodes in the selections which are not roots or leaves are unselect. For all the other, the current selection is extended with complementary nodes or edges.

Finally, finding shortest path find, if it exists, the shortest path between the source node (to be selected on the graph then clicking on “Make selection the source” button) and the target node (to be select on the graph then clicking on “Make selection the target”.

For this version of the tool, the weight for search path is always “1”. Ability of using a property or a function will be provided in future versions. It will also be offered the complementary functionalities related to the management and processing of path objects bringing value to enterprise architects.

Usage example of the shortest path research on the default ArchiMateCG graph

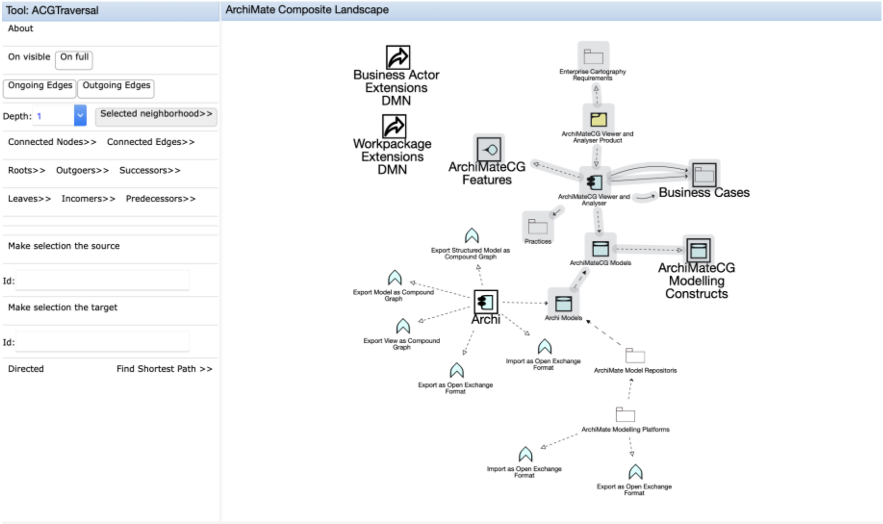
Selecting Ongoing and Outgoing Edges, an clicking when on node is selected (“ArchiMateCG Viewer and Analyser”), and clicking on the “Selected Neighbourhood” button with depth =1, the result is what is shown in the following figure.


Depth: 1

Tool: ACGTraversal
About
On visible
On full
Ongoing Edges
Outgoing Edges
Selected neighborhood>>
Connected Nodes>>
Connected Edges>>
Roots>>
Outgoers>>
Successors>>
Leaves>>
Incomers>>
Predecessors>>
Make selection the source
Id:
Make selection the target
Id:
Directed
Find Shortest Path >>
ArchiMate Composite Landscape
Business Actor Extensions DMN
Workpackage Extensions DMN
Enterprise Cartography Requirements
ArchiMateCG Viewer and Analyser Product
ArchiMateCG Features
ArchiMateCG Viewer and Analyser
Business Cases
Practices
Export Structured Model as Compound Graph
ArchiMateCG Models
ArchiMateCG Modelling Constructs
Export Model as Compound Graph
Archi
Archi Models
Export View as Compound Graph
Import as Open Exchange Format
ArchiMate Model Repositoris
Export as Open Exchange Format
ArchiMate Modelling Platforms
Import as Open Exchange Format
Export as Open Exchange Format

Figure 33: the neighbourhood of a selected node, depth 1

Clicking a second time, it extends the selection providing the neighbourhood of the neighbourhood. The result is the same than using depth =”2” (and so on if clicking n times with n in [1..9] or greater.

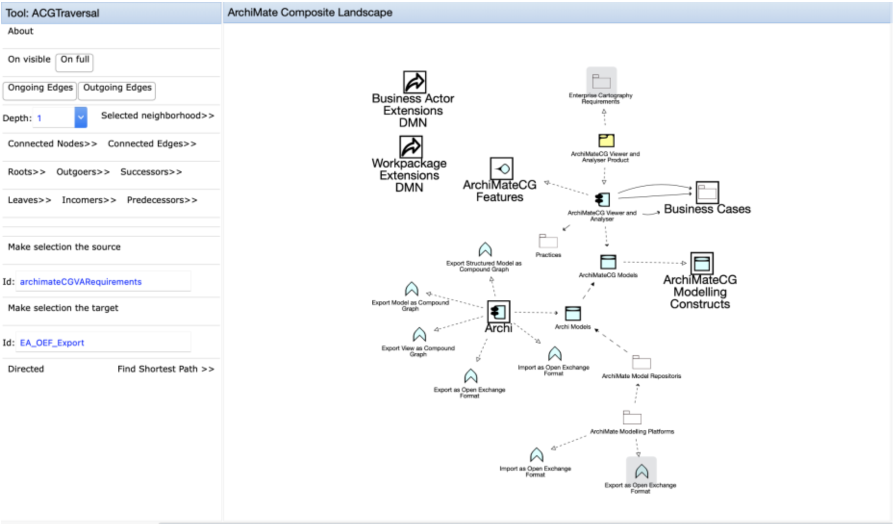

Depth: 1

Tool: ACGTraversal
About
On visible
On full
Ongoing Edges
Outgoing Edges
Selected neighborhood>>
Connected Nodes>>
Connected Edges>>
Roots>>
Outgoers>>
Successors>>
Leaves>>
Incomers>>
Predecessors>>
Make selection the source
Id:
Make selection the target
Id:
Directed
Find Shortest Path >>
ArchiMate Composite Landscape
Business Actor Extensions DMN
Workpackage Extensions DMN
Enterprise Cartography Requirements
ArchiMateCG Viewer and Analyser Product
ArchiMateCG Features
ArchiMateCG Viewer and Analyser
Business Cases
Practices
Export Structured Model as Compound Graph
ArchiMateCG Models
ArchiMateCG Modelling Constructs
Export Model as Compound Graph
Archi
Archi Models
Export View as Compound Graph
Import as Open Exchange Format
ArchiMate Model Repositoris
Export as Open Exchange Format
ArchiMate Modelling Platforms
Import as Open Exchange Format
Export as Open Exchange Format

Figure 34: selection of the next level of neighbourhood

Let’s note that a neighbour is a node. In the resulting selection, neighbour nodes and edges relating the neighbours will be in the selection as well. If “Ongoing Edges” or “Outgoing Edges” is selected, the direction of the relation will be considered in order to select the neighbours.

After selecting the “archimateCGVARequirements” a starting node and the “EA_OEF_Export” node as end (Figure 35), launching the search clicking on the “Find Shortest Path >>” button, the shortest path is first displayed as an ordered list of nodes and arcs (Figure 36). Then after clicking on the button for closing the alert window (name depends on the default language of the desktop environment, here French with “Fermer’, the selected elements of the graph are those of the path (Figure 37).


Tool: ACGTraversal
ArchiMate Composite Landscape
About
On visible
On full
Ongoing Edges
Outgoing Edges
Selected neighborhood>>
Connected Nodes>>
Connected Edges>>
Roots>>
Outgoers>>
Successors>>
Leaves>>
Incomers>>
Predecessors>>
Make selection the source
Make selection the target
Find Shortest Path >>
Business Actor Extensions DMN
Workpackage Extensions DMN
ArchiMateCG Features
Enterprise Cartography Requirements
ArchiMateCG Viewer and Analyser Product
ArchiMateCG Viewer and Analyser
Business Cases
Practices
ArchiMateCG Models
ArchiMateCG Modelling Constructs
Export Structured Model as Compound Graph
Export Model as Compound Graph
Archi
Archi Models
Export View as Compound Graph
Import as Open Exchange Format
Export as Open Exchange Format
ArchiMate Model Repositoris
ArchiMate Modeling Platforms
Import as Open Exchange Format
Export as Open Exchange Format

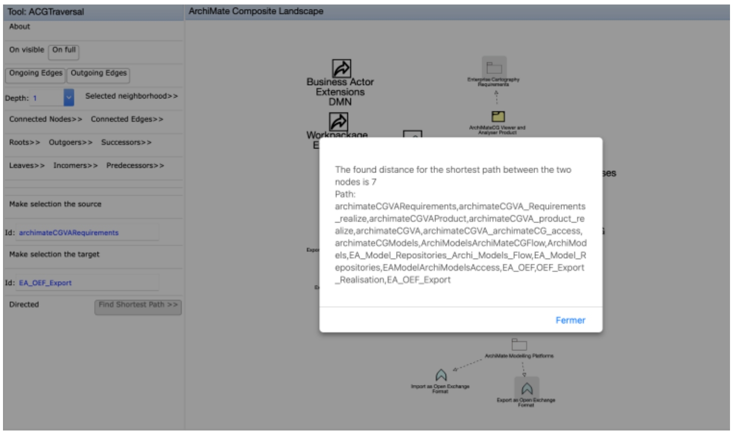
Depth: 1

Id: archimateCGVARequirements

Id: EA_OEF_Export

Directed

Figure 35: the source node and the target node were defined



Depth: 1

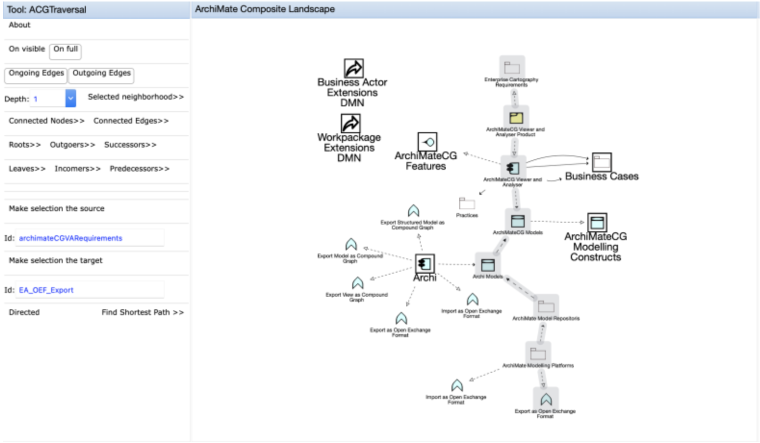
Id: archimateCGVARequirements

Id: EA_OEF_Export

Directed

The found distance for the shortest path between the two nodes is 7
Path:
archimateCGVARequirements,archimateCGVA_Requirements_realize,archimateCGVAProduct,archimateCGVA_product_realize,archimateCGVA,archimateCGVA_archimateCG_access,archimateCGModels,ArchiModelsArchiMateCGFlow,ArchiModels,EA_Model_Repositories_Archi_Models_Flow,EA_Model_Repositories,EAModelArchiModelsAccess,EA_OEF,OEF_Export_Realisation,EA_OEF_Export

Figure 36: Find shortest Path was clicked and the path is displayed



Depth: 1

Id: archimateCGVARequirements

Id: EA_OEF_Export

Directed

Figure 37: the nodes and arcs of the path are selected

Concerning the operations related to Connected Nodes, Connected Edges, Roots, Outgoers, Successors, Leaves, Incomers and Predecessors, it is first required to make a selection. Figure 38 shows the selection of all the visible nodes and edges of the visible graph.

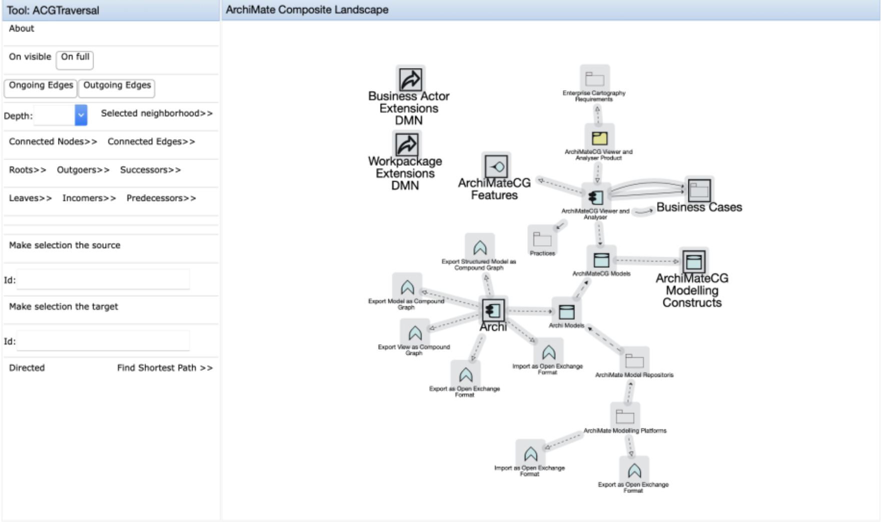

Tool: ACGTraversal
ArchiMate Composite Landscape
About
On visible
On full
Ongoing Edges
Outgoing Edges
Depth:
Selected neighborhood>>
Connected Nodes>>
Connected Edges>>
Roots>>
Outgoers>>
Successors>>
Leaves>>
Incomers>>
Predecessors>>
Make selection the source
Id:
Make selection the target
Id:
Directed
Find Shortest Path >>
Business Actor Extensions DMN
Workpackage Extensions DMN
ArchiMateCG Features
Enterprise Cartography Requirements
ArchiMateCG Viewer and Analyser Product
ArchiMateCG Viewer and Analyser
Business Cases
Practices
Export Structured Model as Compound Graph
ArchiMateCG Models
ArchiMateCG Modelling Constructs
Export Model as Compound Graph
Archi
Archi Models
Export View as Compound Graph
Import as Open Exchange Format
ArchiMate Model Repositoris
Export as Open Exchange Format
ArchiMate Modelling Platforms
Import as Open Exchange Format
Export as Open Exchange Format

Figure 38: all the visible nodes were selected

On the selection of Figure 38, and clicking on Roots, only the root nodes are then selected as illustrated by Figure 39.

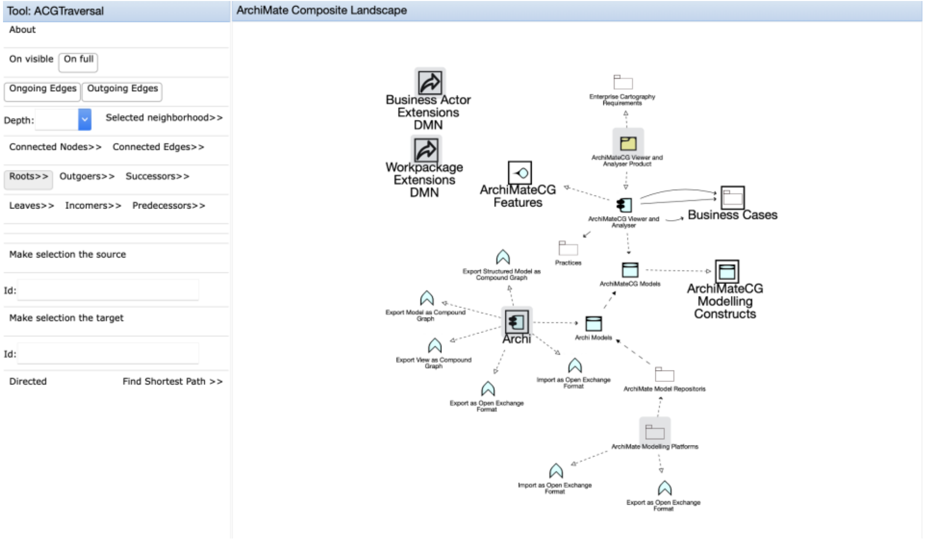

Tool: ACGTraversal
ArchiMate Composite Landscape
About
On visible
On full
Ongoing Edges
Outgoing Edges
Depth:
Selected neighborhood>>
Connected Nodes>>
Connected Edges>>
Roots>>
Outgoers>>
Successors>>
Leaves>>
Incomers>>
Predecessors>>
Make selection the source
Id:
Make selection the target
Id:
Directed
Find Shortest Path >>
Business Actor Extensions DMN
Workpackage Extensions DMN
ArchiMateCG Features
Enterprise Cartography Requirements
ArchiMateCG Viewer and Analyser Product
ArchiMateCG Viewer and Analyser
Business Cases
Practices
Export Structured Model as Compound Graph
ArchiMateCG Models
ArchiMateCG Modelling Constructs
Export Model as Compound Graph
Archi
Archi Models
Export View as Compound Graph
Import as Open Exchange Format
ArchiMate Model Repositoris
Export as Open Exchange Format
ArchiMate Modelling Platforms
Import as Open Exchange Format
Export as Open Exchange Format

Figure 39: Root nodes only are now selected

Let's note that applying such search on any kind of nodes and edges, i.e. ArchiMate constructs, is not always relevant. By hiding nodes and edges by types using the palette (Shift DbClick on types to be hidden), it allows to prepare the appropriate graph with the relevant constructs. However, if you want the search of shortest path to be made on only visible elements, the "On visible graph" button should be selected. If not, hidden part of the graph is considered, and hidden path elements are made visible and selected. Also, it is possible to apply all the operation on the visible nodes only or on all the nodes of the graph, even those not visible by clicking on the buttons "On visible" or "On full" which allow to switch the search mode.

From a practical way, all the operations made available only allows to create a selection of nodes the user will then be able to work on, and nothing more at this stage.

Future features

The underlying Cytoscape library comes with many features related to graph and network based algorithm, such as:

- Finding shorter path or optimal paths between nodes using properties or algorithms
- Usage of traversal graph algorithms in order calculating weights for nodes and arcs, based on appropriate metrics, eventually relying on existence of paths, with some criteria on distances + efficient restitution means with appropriate visualization technics (coloured graphs, size of nodes and width of edges, symbols, etc. applied to interactive graphs or matrixes)
- Clustering for grouping similar nodes and make emerge a simple graph

This can be very valuable for change impact analysis, detection of anomalies, model analytics, etc.

So new “tools” will be created and made available through the tool pane, selectable in a dedicated menu, driven by users’ request. Let’s note that one of the underlying drivers of ArchiMateCG is to make emerging innovative high added value features taking advantage of Composite Graph Visualisation and algorithms for supporting Architects and Enterprise Architecture Model managers.

An empty tool was made available for this version, called ACGAnalytics (for ArchiMate Compound Graphs analytics).

Usage related to data aggregation and end to end processes

ArchiMateCG have been developed having in mind the ability to aggregate various data assets distributed in various, heterogeneous and siloes legacy Enterprise data and model repositories.

ArchiMate as Open standard and as de facto standard was selected for its ability to provide a skeleton for aggregating data coming from the various stakeholders covering each a part of the spectrum of enterprise architecture: enterprise motivation, business models, information system models, information and technologies realising the application of the information system and transformation plan, being roadmaps with portfolios of projects, or single complex project or programs involving all these stakeholders.

In addition, a specification exists for an open exchange format for ArchiMate model, which is supported by many software products supporting ArchiMate modelling.

As ArchiMate is a relatively simple language, it is also very easy developing some import and export scripting, relying on various data syntax, being XML with XSD schemas, JSON with or without schemas, RDF and OWL with various syntaxes and schemas derived from ArchiMate, or eventually XMI files.

The goal here is to support architecture analysis after grabbing many various distributed data and to be then able to perform an analysis, and to publish results for import on different targeted tools.

The stakeholders are Processes, Methods and Tools departments, Information System departments, Quality departments, Human Resources departments, Programs' Operational managers, etc.

For the current version:

- Two export/import chains have been implemented based on JSON syntax and creation of import and export scripts: jArchiCG from Archi, Cytogen from Enterprise Architect… (The list could vary as the importer and exporter are developed externally, being based on open standards or on specific flows between applications of a specific enterprise environment. In next versions, the available exports will not be hardcoded anymore, but parameterized)
- One import relying on the Open Exchange Format for ArchiMate.
- One export of a selection as a set of 3 zipped CSV files

ArchiMateCG and Semantic Web Ontology

Introduction: the motivation

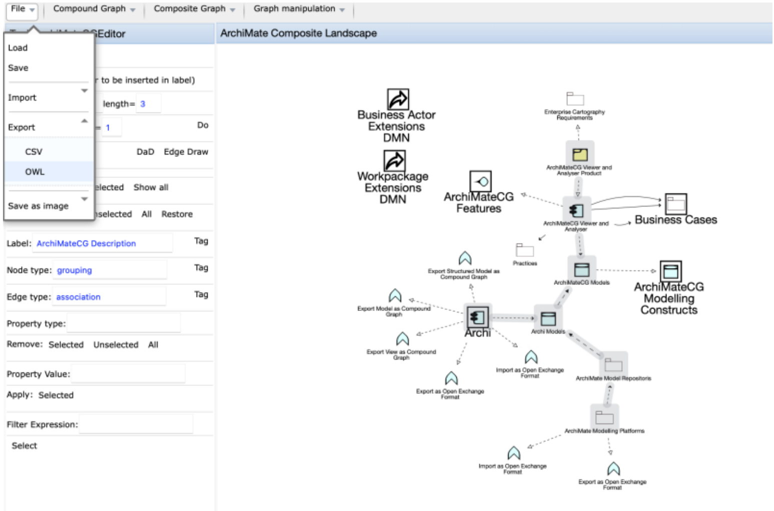
Using the OWL export produces an OWL representation of the selection relying on the JSON LD syntax. It can then be used in many ways:

- As semantic annotations of an HTML document according to Linked Data practices
- As a set of individuals to be used commonly with an OWL representation derived from ArchiMate, which will allow to take advantage of logical validation, inferencing of links (typing, object properties, etc.) based on rules related to ArchiMate and/or considered extensions/complementary ontologies (with mapping rules, based on equivalence and subsumption). The potential intended usages with value creation are:
    - logical model validation,
    - automated completion of models based on rules and inferences created by inference engine
    - Semantic validation of various extensions based on sets of specialization, object/data properties and associated rules.
    - Translation from ArchiMate to other languages based on semantic equivalences and subsumptions.
    - Semantic aggregations of multiple domain languages, including those which specialize subset of the ArchiMate language (e.g. BPMN for business processes)

Usage with Protégé standalone (i.e. without a predefined ontology derived from the ArchiMate language)

1- Loading
2- Navigating
3- Graph visualization and querying with Ontograph
4- Querying with DL queries

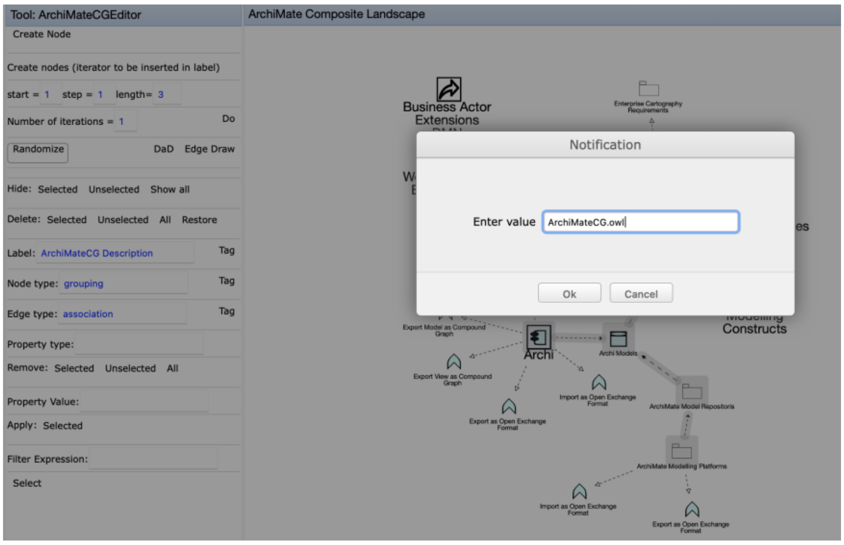
As illustration, the ArchiMateCG graph model opened when starting ArchiMateCG, which provides an ArchiMateCG architectural description of ArchiMateCG itself, was exported in OWL.



Label: ArchiMateCG Description
Node type: grouping
Edge type: association

File
Compound Graph
Composite Graph
Graph manipulation
Load
Save
Import
Export
CSV
OWL
Save as image
GEditor
r to be inserted in label)
length= 3
= 1
Do
DaD
Edge Draw
elected
Show all
nselected
All
Restore
Tag
Tag
Tag
Property type:
Remove: Selected Unselected All
Property Value:
Apply: Selected
Filter Expression:
Select
ArchiMate Composite Landscape
Business Actor Extensions DMN
Workpackage Extensions DMN
ArchiMateCG Features
Enterprise Cartography Requirements
ArchiMateCG Viewer and Analyser Product
ArchiMateCG Viewer and Analyser
Business Cases
Practices
Export Structured Model as Compound Graph
ArchiMateCG Models
ArchiMateCG Modelling Constructs
Export Model as Compound Graph
Archi
Archi Models
Export View as Compound Graph
Import as Open Exchange Format
Export as Open Exchange Format
ArchiMate Model Repositoris
ArchiMate Modelling Platforms
Import as Open Exchange Format
Export as Open Exchange Format

Figure 40: OWL export menu selection



Tool: ArchiMateCGEditor
Create Node
Create nodes (iterator to be inserted in label)
Do
Randomize
DaD
Edge Draw
Hide: Selected Unselected Show all
Delete: Selected Unselected All Restore
Tag
Tag
Tag
Remove: Selected Unselected All
Apply: Selected
Select
ArchiMate Composite Landscape
Business Actor Extensions
Enterprise Cartography Requirements
Modelling Constructs
Export Model as Compound Graph
Archi
Archi Models
Export View as Compound Graph
Import as Open Exchange Format
Export as Open Exchange Format
ArchiMate Model Repositoris
ArchiMate Modelling Platforms
Import as Open Exchange Format
Export as Open Exchange Format
Notification
Ok
Cancel

start = 1 step = 1 length= 3

Number of iterations = 1

Label: ArchiMateCG Description

Node type: grouping

Edge type: association

Property type:

Property Value:

Filter Expression:

Enter value ArchiMateCG.owl

Figure 41: Giving the name of the exported OWL file

Once the file saved, you can open it with Protégé, which correspond to the loading.

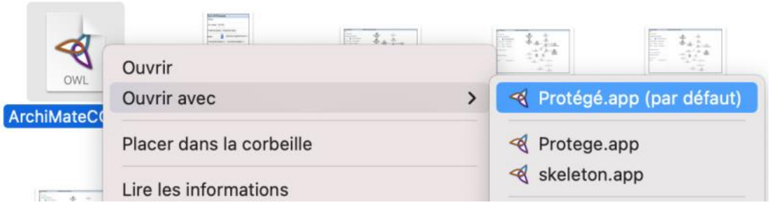
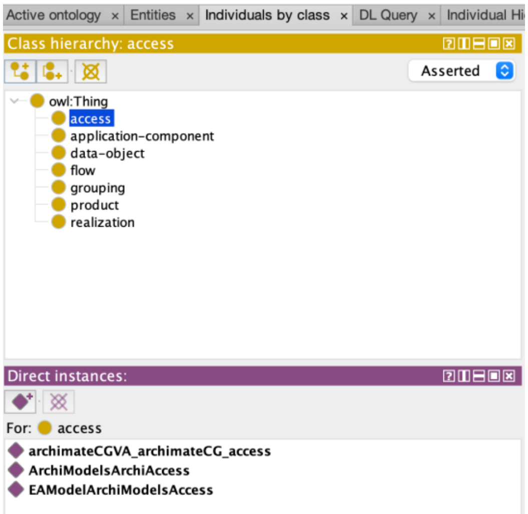


Figure 42: Opening the exported file with Protégé



For: access

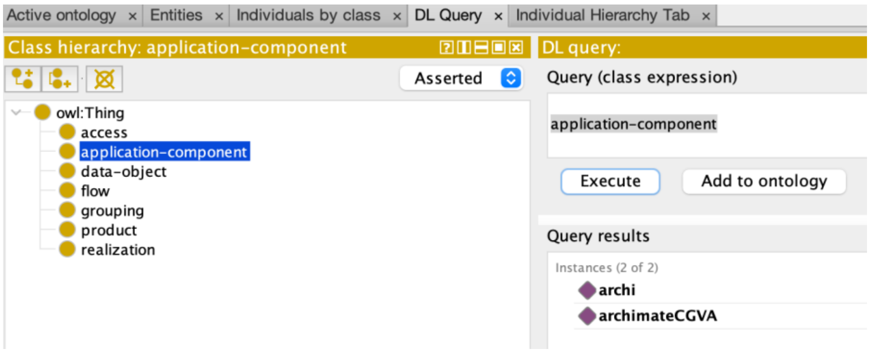
archimateCGVA_archimateCG_access

ArchiModelsArchiAccess

EAModelArchiModelsAccess

Figure 43: the model elements are loaded, and class are automatically created

It is then possible to take advantage of Protégé functionalities, related to reasoning (by mean of reasoning engine plugins, e.g. the one based on Pellet) or visualisation of an ontology as a graph (by mean of Ontograph Protégé plugin). Descriptive Logic queries (or DL queries) as enabled by the usage of a reasoning engine is illustrated by Figure 44. Visualisation with Ontograph is illustrated by



application-component

archi
archimateCGVA

Figure 44: DL query on the exported model

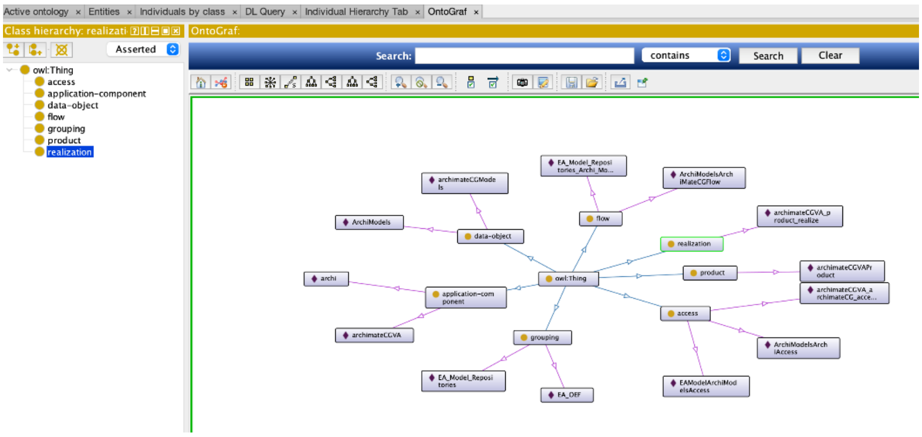


Figure 45: visualisation of the model with Ontograph

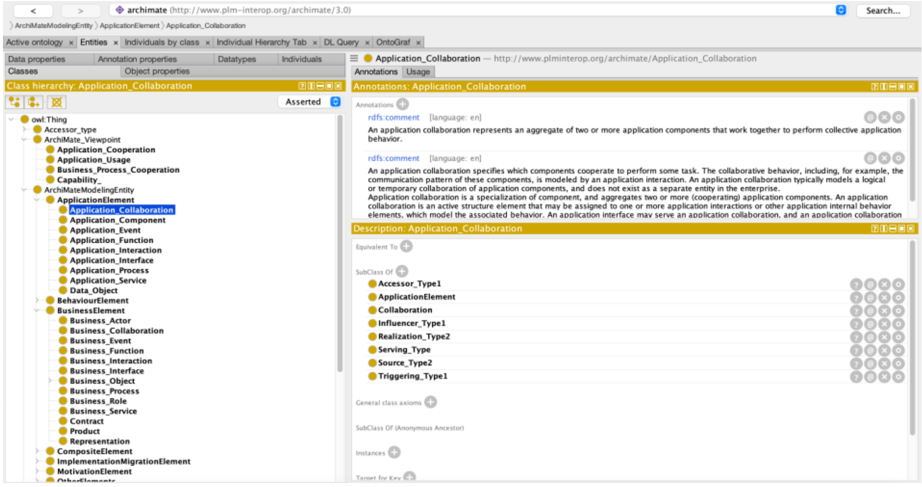
Here we just have the data corresponding to a given model, but don't have all the description of the ArchiMate language consisting in an ontology, i.e. a set of OWL classes, Properties and rules describing the language constructs. Having such a description will leverage usage of the model as an ontology, by providing many rules allowing to validate the model, or to ensure complementary rule based inferences enabled by the reasoning engine.

Usage with Protégé with a predefined ontology (i.e. with a predefined ontology derived from the ArchiMate language).

1- Loading
2- Launching the reasoner
3- Querying with DL Queries
4- Exporting inferenced in a new model
5- Loading the completed model
6- Navigating
7- Graph visualization and querying with Ontograph
8- Querying with DL queries or SparQL

Figure 40 illustrates what an ArchiMate definition in OWL could looks like. They can be more or less complete, including or not structural and semantic rules, including or not the definition of the language constructs, containing or not ArchiMate concepts which are not part of the visual language, or finally being formalized to such or such OWL profile, each profile proposed by OWL being dedicated to such or such family of technologies with different performances, but also with specific constraints on the

OWL language constructs to be used. Let’s consider OWL specifications or tutorials for knowing more about it.



rdfs:comment [language: en]

An application collaboration represents an aggregate of two or more application components that work together to perform collective application behavior.

rdfs:comment [language: en]

An application collaboration specifies which components cooperate to perform some task. The collaborative behavior, including, for example, the communication pattern of these components, is modeled by an application interaction. An application collaboration typically models a logical or temporary collaboration of application components, and does not exist as a separate entity in the enterprise.
Application collaboration is a specialization of component, and aggregates two or more (cooperating) application components. An application collaboration is an active structure element that may be assigned to one or more application interactions or other application internal behavior elements, which model the associated behavior. An application interface may serve an application collaboration, and an application collaboration

Figure 46: ArchiMate definition in OWL

The ArchiMate ontologies provided with ArchiMateCG

Several formalizations of ArchiMate in OWL have been produced over the time with various purpose. Email contact address: Nicolas.figay@airbus.com

Perspectives of evolution for ArchiMateCG

- Usage taking advantage of inferencing (make some implicit links explicit based on rules and using reasoning, before to apply traversal of graphs for calculating weights, distances, etc. It implies that it should be possible to import OWL files containing ArchiMate constructs typed individuals in ArchiMateCG
- Decorating a compound graph with object properties defined by ontologies…

Exploring time aspect with Enterprise Architecture and ArchiMateCG

The question arises concerning how to deal with time when working with ArchiMate, as it is all about support continuous transformation of the enterprise, relying on a set of strategic roadmaps and derived portfolio of projects for their realizations. The work-package ArchiMate modelling construct is the one allowing to capture elements which can be included in a planning, with start and end dates, planned or actual. Then the “Plateau” ArchiMate modelling construct is about the construction of a persistent boundary per project, stating what the starting situation for the considered working landscape elements is, and what the targeted situation is. Then some work can be elaborated having in mind what is expected and why. Because the plateaus are defined within the whole enterprise landscape, it allows to identify all the stakeholders and all the required contributors to the success of the project, and the complete environment where each plateau stands, with interfaces and containments. So information about roadmaps can also be reflected on timelines, which should encompasses all the related projects. Finally, we can capture time related information for any item composing the whole landscape of the enterprise, such as effectivity per date or applicability per date.

In ArchiMateCG, timeline tool is about the exploration of what can be done coupling the dynamic and interactive compound graph realizing ab Architectural Description using the ArchiMate visual modelling language, and a dynamic interactive timeline feature where we can represent all the time related elements of the model, in a way it allows to create value by better visualizing and analysing the temporal aspect of enterprise transformations.

The questions which can be raised (visual mining) and analysis which can be performed (graph analytics) are various and numerous.

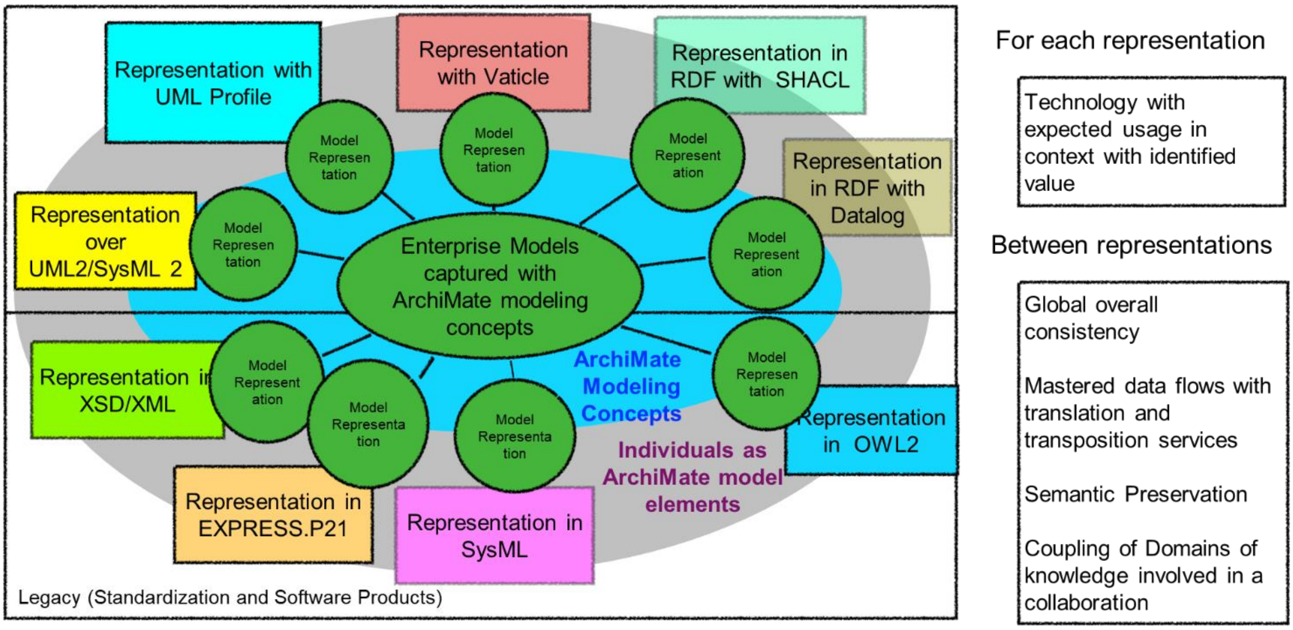
The Timeline tool aims at exploring the potential of associating these features in order to create value for the architects, supporting the needs for their activities, but also any other stakeholders which can take benefits of exploring the landscape where they stand, taking into consideration its continuous evolution and transformation for synchronising activities.

Future Features

If only few imports and exports were made available so far, it is planned to continuously extend them, relying on state of the art innovations in terms of data exchange and aggregation and value creation.

The current backlog contains:

- Import of OWL for decorating the ArchiMateCG compound graphs or enriching them with inferenced relations.
- Export relying on Open Exchange Format for ArchiMate
- Import/Export in XMI considering the ArchiMate representation over UML2/SysML defined with the Polyglot hypermodel for Interoperability
- Import/Export for Vaticle considering the ArchiMate representation over hypergraph defined with the Polyglot hypermodel for Interoperability
- Import of schemas for adding properties sets to the graph
- Integration of object properties with alternative graph representation displaying typed links and not relationships.
- Import of ontologies for enrichment of the graph element properties and specialisations relying on legacy taxonomies.
- Creating a visual mode allowing to display object properties as edges, while ArchiMate relationships will be represented as nodes with edges pointing source and target, and edge create from a node to its parent.
- A new tool for visual group representations which are not implying partitioning
- An animation tool for storytelling and demonstration scenarios
- Dedicated algorithms for shortest path responding to Architect specific questions


Representation with UML Profile
Representation with Vaticle
Representation in RDF with SHACL
Representation in RDF with Datalog
Representation over UML2/SysML 2
Enterprise Models captured with ArchiMate modeling concepts
Model Representation
Model Representation
Model Representation
Model Representation
Model Representation
Model Representation
Model Representation
Model Representation
Model Representation
Representation in XSD/XML
ArchiMate Modeling Concepts
Individuals as ArchiMate model elements
Representation in OWL2
Representation in EXPRESS.P21
Representation in SysML
Legacy (Standardization and Software Products)
For each representation
Technology with expected usage in context with identified value
Between representations
Global overall consistency
Mastered data flows with translation and transposition services
Semantic Preservation
Coupling of Domains of knowledge involved in a collaboration

Advanced visualisation: present and future

The current version of ArchiMateCG integrated some advanced features related to DataViz based on the emerging Web standards for Navigators: combination of SVG, HTML, DOM and JavaScript, as performed by Cytoscape.js, but also by various libraries such as D3.js

This creates many opportunities for future innovative features in terms of semantic cartographies (the same underlying model with clearly established semantic and advanced interactive visualisation techniques) for ArchiMateCG, which will be introduced in an opportunistic way, driven by value creation and innovation.

It includes:

- Projection of a graph subset on a radar chart (Figure 47)
- Projection of a graph subset on SWOT (Strength, Weakness, Opportunities and Threats)
- Alternative visualisation on interactive timelines[3] (Figure 48)
- Alternative visualisation relying on interactive Treemaps (Figure 49)
- Interactive matrixes, extended for taking into account matrixes (Figure 50)
- And many others.

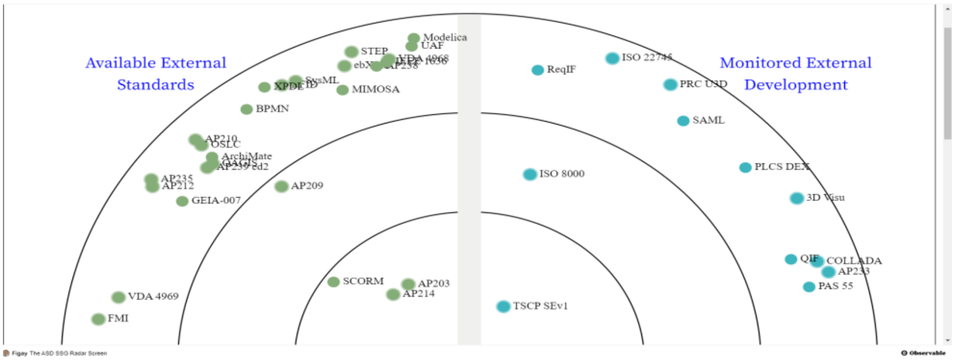
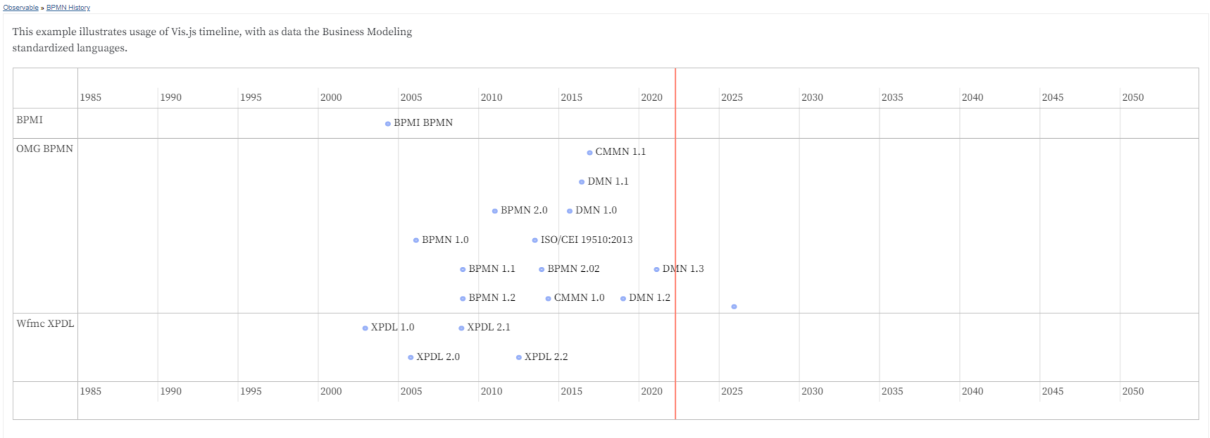


Figure 47: Radar chart example


Observable » BPMN History

This example illustrates usage of Vis.js timeline, with as data the Business Modeling standardized languages.

Figure 48: Timeline example

[3] Let's note that timelines uses can now be explore with the timeline tool

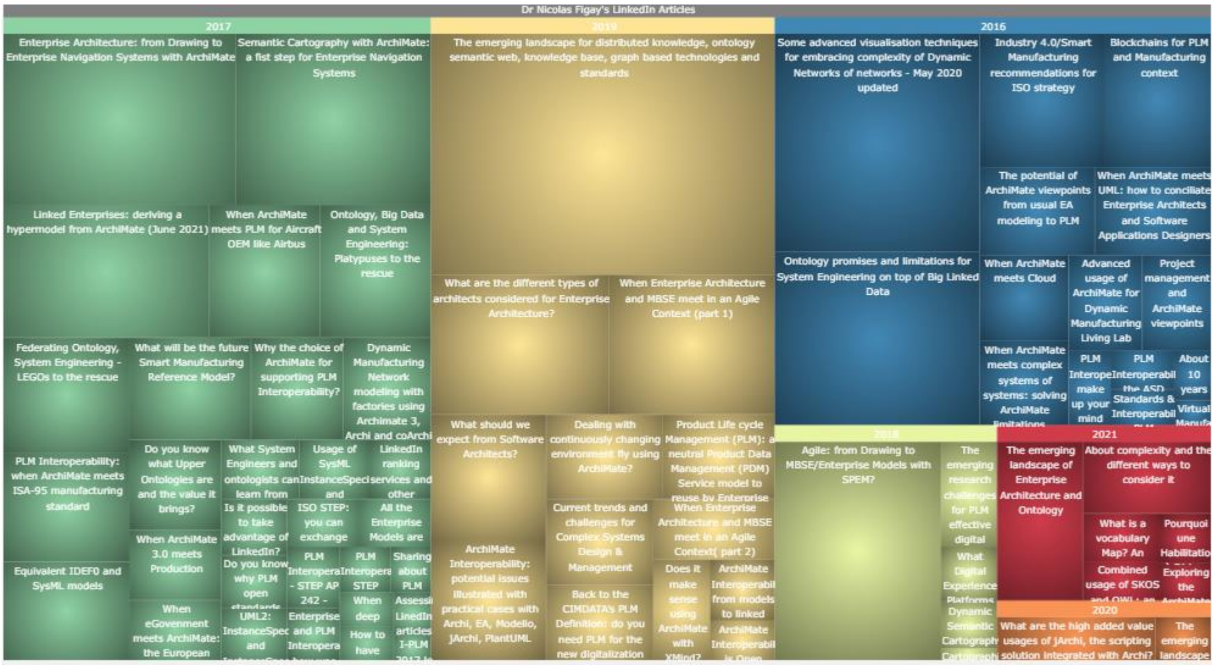
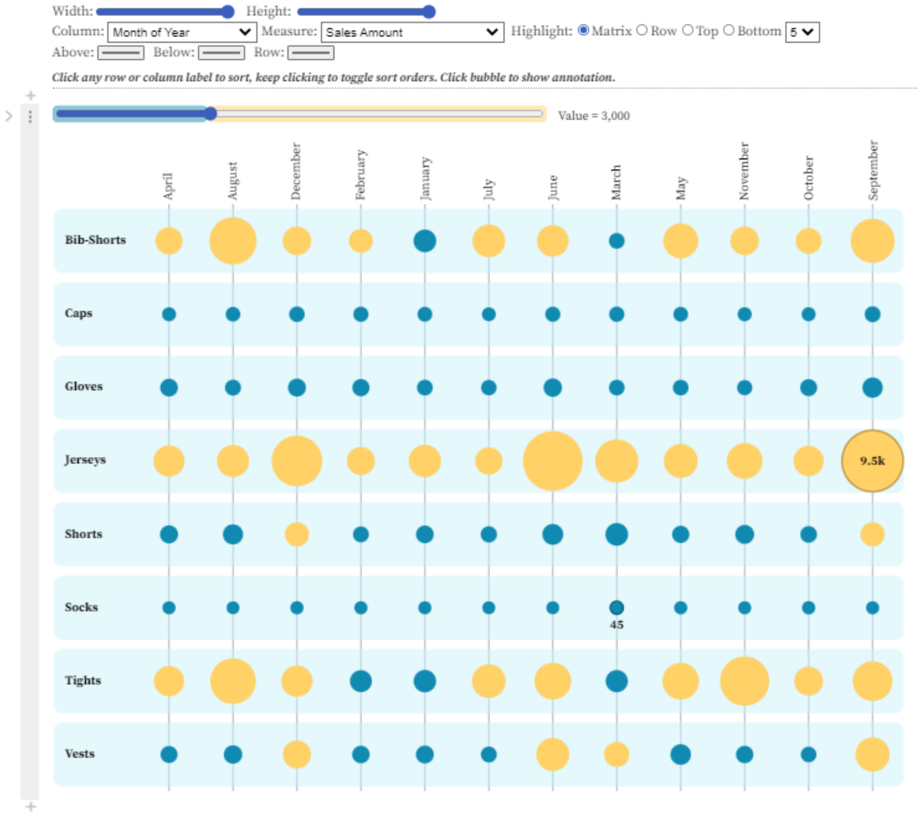


Figure 49:Treemaps illustration



Width: Height:
Column: Month of Year Measure: Sales Amount Highlight: ● Matrix ○ Row ○ Top ○ Bottom 5
Above: Below: Row:

Click any row or column label to sort, keep clicking to toggle sort orders. Click bubble to show annotation.

Value = 3,000

Ch

Figure 50: Matrix representation of a graph illustration

Animations for storytelling and demonstration scenarios

This is one of the functionality which can be envisaged for future versions of ArchiMateCG.

It could relies on the advanced visualisation functionalities combined with graph visualisation and analytics libraries based on scripting which are provided with Cytoscape.js.

Figure 51 illustrates it (this is a gif image you can save in order to see the animation).

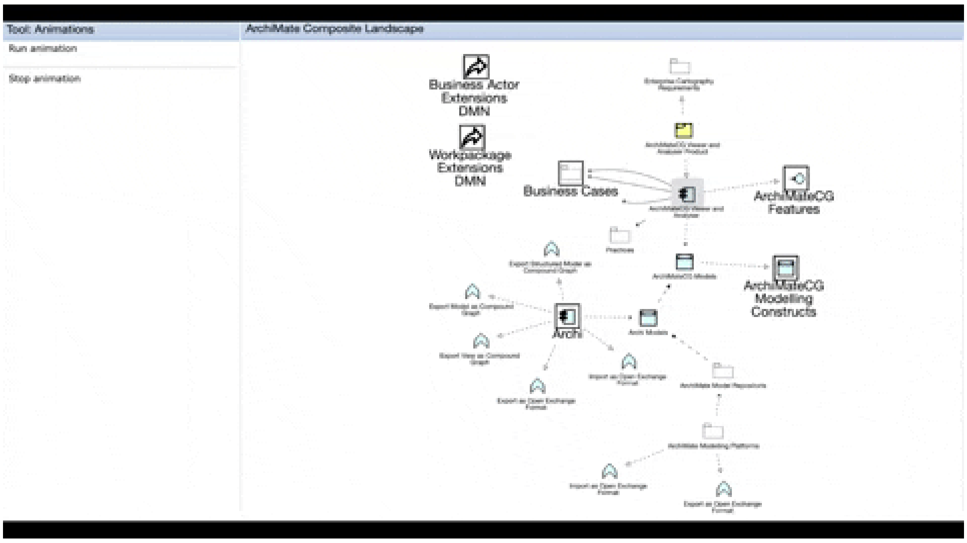

Tool: Animations
Run animation
Stop animation
ArchiMate Composite Landscape
Business Actor Extensions DMN
Workpackage Extensions DMN
Business Cases
ArchiMateCG Features
Practices
ArchiMateCG Modelling Constructs
Archi
Archi Models
ArchiMate Model Repositories
ArchiMate Modeling Platforms
Export Model as Compound Graph
Export View as Compound Graph
Export as Open Exchange Format
Import as Open Exchange Format

Figure 51: GIF dynamic image illustrating an animation

The proposed animation is quite trivial and has definitively not any interest except showing an animation which can be run in an actual model, which required only 2 or 3 lines of scripting: all the nodes to be put at the same location with a smooth animation having within a given duration.

```
cy.nodes().forEach(function( node, i ){
    node.animate ({position: {x: 100, y: 100}}, {duration: i*1000})});
```

The data associated to each model element could be eventually changed and the same animation script applied to it, being MS PowerPoint animation or MS Excel macro similar. It will of course depends on the nature of the automated operations and to their adherence to specific and explicit data their descriptions require. E.g. if you specify a precise positioning for each node with explicitly given coordinates, there will be an issue if you try to position a node that doesn't exist or isn't displayed.

The idea is also to prevent coding when possible when willing to create animations, as not all the targeted ArchiMateCG users are developers.

A demonstration tool, "Animation" is associated with this version of ArchiMateCG. It is really a pre-alpha version, and it is just a hook for future potential features related to animations creating values for the architectural description supported by ArchiMateCG.

In order to have an idea on how it can contribute to explaining model based approaches relying on ArchiMate for deploying new capabilities contributing to end to end process, you can have a look at the video available here (restricted to Airbus).

Similar demonstration will be made available soon with open access.

Visual group representations which are not implying partitioning

Compound graphs are quite relevant when willing to represent hierarchies, each node having at the maximum one parent node. It is suited for representing containments and composition relationships. But how to deal for representing multiple categorisations simultaneously or aggregation? Compound nodes as implemented by the used graph libraries is not suited for such a purpose.

A Cytoscape plugin exists, called "bubblesets", which allows to define and visually represents sets which are grouping several nodes and can intersect with other sets. The usage and integration is being studied and assessed in order to be integrated with ArchiMateCG. At this stage, several issues have been identified and are being addressed, in order to be able to make some proposal which will then will have to be assessed in terms of value creation for the ArchiMateCG users, in particular the architects. The two following figures (gif files you can save on your disk to see the animation) are showing how it allows drawing sets 1) with the ability to aggregate them using drag and drop 2) combining compound nodes and bubble sets 3) with two sets having an intersection.

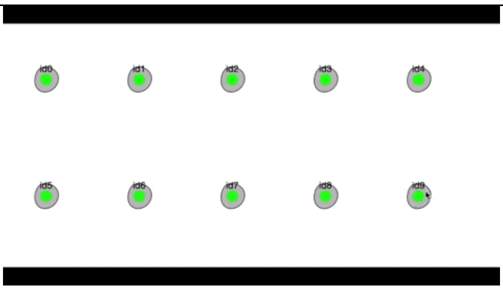


Figure 52: Bubble sets which can be aggregated using drag & drop (GIF)

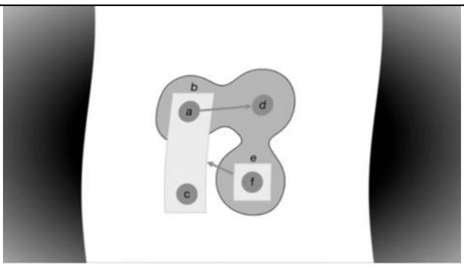


Figure 53: Bubble Set combined with compound nodes

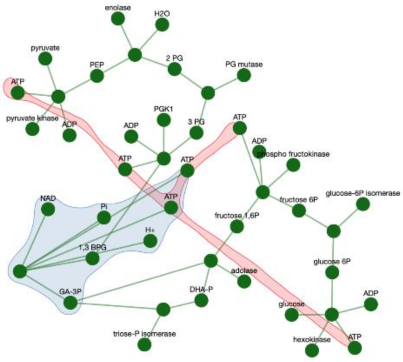


Figure 54: Two intersecting bubble sets

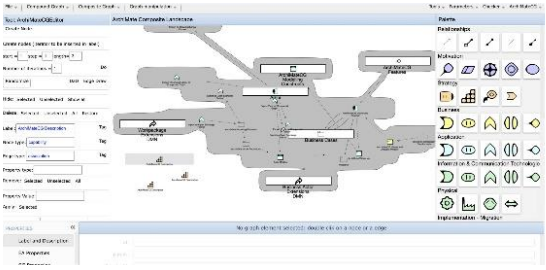


Figure 55: First integration test in ArchiMateCG

Note: In parallel, there is assessment of other open source and free libraries proposing similar feature. On library in Python, NetworkX, with a port on JavaScript which is unfortunately partial and without the expected feature. What can also investigated is Venn diagrams drawing solutions such as Interactivenn. The following figure shows what can be represented with those tools.

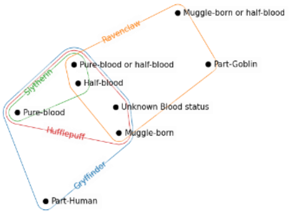


Figure 56: NetworkX hypergraph drawing

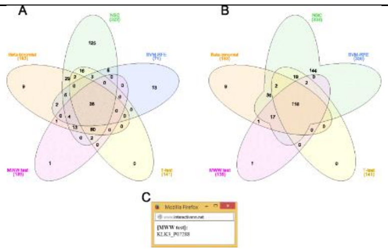


Figure 57:Interactive Venn illustration

Note: it should be consider that others ways could be envisaged for aggregation visualisation.

Quantitative analysis of enterprise architecture: present and future

At this stage, ArchiMateCG is enabling basic features for quantitative analysis, mainly what is related to graph analytics with calculation of degrees of a usual graph, using the tool ACG Degrees.

In addition, the ACG Traversal tool is proposing the shortest path between two nodes, considering the number of arcs of the path.

However, quite more can be envisage of exploiting a graph with attached properties or values which are graphs related (such as degrees) which can be used in order to calculate a weight for nodes or edges, and to use them in order to calculate a distance which will be used for calculating a shortest path.

This can be extended considering that ArchiMateCG is dealing with compound graphs, e.g. with a hierarchical organisation of graphs with parent and child nodes. We are here consequently working here with modular models, with nested elements creating a partitioning. It means that values qualifying graphs can be extended, as the different kind of pathways. This is illustrated in [1], for which an adaptation was created and published for Cytoscape for generic usage (cf. https://github.com/iVis-at-Bilkent/cytoscape.js-graph-algos)

Finally, the semantic of the used ArchiMate constructs, with concerns and quantities suited for Enterprise architects, should also be considered, in order to respond to precise and specific questions which are relevant for architects and create values. It is in particular true for measuring the work required from a current as is architecture and some potential targeted alternative architectures, with different strategies and work plans. Different uses we can imagine can be derived from current practices, such those defined in the provided references.

References

1. U. Dogrusoz, A. Cetintas, E. Demir and O. Babur, "Algorithms for effective querying of compound graph-based pathway databases", BMC Bioinformatics, 10(1), pp. 1-16, 2009Singh, P.M., van Sinderen, M.J. (2015). Lightweight Metrics for Enterprise Architecture Analysis. In: Abramowicz, W. (eds) Business Information Systems Workshops. BIS 2015. Lecture Notes in Business Information Processing, vol 228. Springer, Cham. https://doi.org/10.1007/978-3-319-26762-3_11
2. Iacob, ME., Jonkers, H. (2006). Quantitative Analysis of Enterprise Architectures. In: Konstantas, D., Bourrières, JP., Léonard, M., Boudjlida, N. (eds) Interoperability of Enterprise Software and Applications. Springer, London. https://doi.org/10.1007/1-84628-152-0_22
3. Raouf Khayami, Qualitative characteristics of enterprise architecture, Procedia Computer Science, Volume 3,2011,Pages 1277-1282,ISSN 1877-0509, https://doi.org/10.1016/j.procs.2011.01.004.(https://www.sciencedirect.com/science/article/pii/S1877050911000056) Abstract: Constant changes in information technology (IT) and business environments have made the demand for a powerful management for IT systems more pressing. Enterprise architecture is a framework to develop and maintain IT, to achieve organizational goals and to manage resources of this technology. Enterprise Architecture (EA) quality is a multi-dimensional content which is not easily distinguishable and measurable. To determine this content more exact, the qualitative models have been presented in which different aspects of this matter are investigated. This paper attempts to introduce mentioned about determine EA qualification and its qualitative characteristics more clearly. This article can be used as a reference to investigate EA qualification and its models. Also, it can help stakeholders to explain the qualitative requirements more exactly.

Keywords: Quality model of enterprise architecture; Enterprise architecture characteristics; Enterprise architecture, Quantitative Alignment of Enterprise Architectures with the Business Model

4. Engelsman, Wilco & Wieringa, Roel & van Sinderen, Marten & Gordijn, Jaap & Haaker, Timber. (2020). Transforming e 3 value models into ArchiMate diagrams. 10.1109/EDOC49727.2020.00012. Bridging value modelling to ArchiMate via transaction modelling
5. Iacob, Maria-Eugenia & Meertens, L. & Jonkers, Henk & Quartel, Dick & Nieuwenhuis, Bart. (2012). From enterprise architecture to business models and back. Software & Systems Modeling. 13. 10.1007/s10270-012-0304-6. Cross-layer Enterprise Architecture Evaluation: An Approach to Improve the Evaluation of TO-BE Enterprise Architecture
6. Dietz, Jan L. G.. "Understanding and Modelling Business Processes with DEMO." ER (1999).https://sparxsystems.com/enterprise_architect_user_guide/15.2/guidebooks/tech_estimation.html
7. Florez, H., Sánchez, M. & Villalobos, J. A catalog of automated analysis methods for enterprise models. SpringerPlus 5, 406 (2016). https://doi.org/10.1186/s40064-016-2032-9
8. https://numberdyslexia.com/graph-theory-applications-in-real-life/
9. https://www.javatpoint.com/graph-theory-applications
10. https://www.masterclass.com/articles/graph-theory
11. http://www.hoonzis.com/applications-of-graph-theory/
12. https://www.dharwadker.org/pirzada/applications/
13. https://marceaucoupechoux.wp.imt.fr/files/2018/02/graphtheory.pdf

Cartography

The description of an architecture is not only constituted by model elements and relationships (the model) but also a set of views which are being provided following a process formalizing the applied practice. Views not only filtering model elements, but are constituting by themselves an aggregate of information, provided for some stakeholders, and associated to a purpose and to a concern (as defined in ISO 421010). A view is constituting an entry point to the model. However, we can't just deal with an unstructured set of views, we should be able to structure them as a model with a coarse grains modularity, which can be displayed in order to:

- Reflect some organic structuration which can be navigate with a proposed navigation path for given stakeholders
- Reflect the applied transformation process, links between data assets needed for the views to be produced and the dedicated activities, and who is responsible for such or such view
- …

In fact, many ways to create "views on views" can be imagine, for various purposes, concerns and stakeholders.

In some extends, we can consider that a view is an aggregate of model elements represented visually, and that it can correspond to any aggregation which can be captured in the model itself, defined extensively, and which can be created manually or through the response to a query. Such a query can be formalized logically relying on natural language, but on a computable language as well, in particular by relying on descriptive logic.

So a planned tool for ArchiMateCG is dedicated to the building of structured views and the way to logically defined them.

IDentity cards

Not all people are used to complex composite graphs based on complex models, and it is sometimes difficult to isolate a model element from its environment. It could be great, for each element, to create a synthetic description constituting all the main information which is to be known about the element, guiding and ensuring the completeness of what is to be defined by this element, and managed by the element owner with support of all required responsible and contributors.

To some extent, an ID card is a formalization of a view on the model element, with a particular presentation, a dedicated form on the screen of your PC, or a document template. In addition to the informational content, you may have some metadata concerning the "ID Card" itself, which will be a managed document.

However, some attention is to be paid not only relying on model elements, but also on relationships between them and between the views. So an "ID Card" could also be considered for a set of elements and some particular compound networks they are constituting. This can be a trap for an "ID Card", only to show some details about an element, but not providing a holistic view which can be easily and dynamically navigated. In the reverse, if viewing the whole information network, a user can be a little bit lost when having to filter and to provide a synthetic managed representation of the considered element collecting the most important information to be considered and managed.

So how to combine a holistic view as a compound graph and ID card creation and management? ArchiMateCG will propose a tool to automatically generate the ID card from the graph. In the reverse, it will be possible to modify the graph from the completion or change of the ID Card, as a reference source of information

Visual polyglot model element

If symbols can be used to indicate the type of a model element, other categorizations so types can be applied to any model elements other than the used modeling construct. So the question is how to be able to reflect visually on the produced graph the multiple categorizations which can be applied to a model element. The idea here will be to propose new way of structuring and displaying graphs, relying on hypergraphs dedicated to such multi typing coming from multiple categorizations, relying on research works realized on:

- Polyglot hypermodel for interoperability
- Innovative visualization proposed by legacy tools (e.g. some mindmap solutions)

Some open questions in terms of research

Usage and limitations of compound graphs

- Experimentations and prototyping raised several questions concerning how to deal with compound graphs, such as:
- Distinction of edges representing actual ArchiMate relationships and edges created from collapsing actions on the graph (cf. Limitations)
- Creation of collapsible matrixes representing compound graphs
- Relevance of relations between nodes which are not at the same level of decomposition for an architectural representation of an enterprise. What should be allowed or not, and to enforce or check it for working on valid models?
- How to deal with objects associated to the boundary of an element, e.g. entry points, as supported by languages such as UML2 and SysML? It is relevant in particular when having to define a path implying to “enter in” and to “go out of” the considered compound element. E.g. entry points where a service can be accessed for an application.

Semantic cartography

- Alignment between the structure of visualization means and underlying semantic and data structure.
- Enterprise landscape analytics and associated relevant dynamic interactive visualizations.
- What should be the invariant part of a landscape, where various information are projected?
- What should be the role of visual groups if used in addition to nesting?
- Can we apply the approach to something else than Enterprise Architecture and ArchiMate?

Enterprise Architecture

- What is the value creation for enterprise architecture practices and associated processes?
- Semantic Enterprise Cartography for people, teams or the whole organization? Process of data acquisition with the accurate quality, i.e. accuracy and veracity?

Interoperability

- How to model with ArchiMate the context where interoperability must occur, for end to end applications supporting end to end collaborations between collaborating enterprises with their own internal processes and applications realizations and infrastructures?
- How to ensure flows between ArchiMateCG and the different enterprise data repositories without loss of information?
- Interchange of a model represented by simple flat graphs (LPG), compound graphs (Cytoscape) and system modeling languages? E.g. it was identified that an ArchiMate relation between a model element and a relation can’t be represented by a node or an edge of a graph. It was also identified that unlike SysML or UML2, it is not possible to define model elements associated to a boundary. Can we elaborate on the practices for building interoperability and preventing silos from the analysis of the encountered problems, e.g. providing some patterns and ant-patterns