

Labnaf Change History

B0701

Change Element Type

You can now retain all properties and values, even if they are not applicable to the new element type.

Change Element Type

Current Element Type

Application Product

Current Detailed Type

Component/LABN::LABN_ApplicationProduct

Use Type of Selected Element

Swap

New Element Type

Application

New Detailed Type

Component/LABN::LABN_Application

Use Type of Selected Element

Keep current element properties and values

Scope: Where to search for elements with type matching Current Type?

Selected elements or packages Full Package Hierarchy

Elements on active diagram

For Administrators only

All elements in repository Include Configuration Elements

Preview Changes Ask confirmation

Change Type

Help

Close

Generate List Of Element Types

Reporting

You can generate a tabular report on selected elements which are not of the same type:

- Select elements in a diagram or in the project browser
- Select Generate Tabular Report (now, you can still change your element selection if you want)
- Select Custom Report and select “Any” as element type

The properties that you can include are common to any element type, since the selected elements to be reported are not of the same type.

Custom Report

Select an element type

- Any
- Access Point
- Activity
- Application
- Application As A Service
- Application Component
- Application Deployment Set
- Application Function
- Application Group
- Application Platform
- Application Product
- Application Service
- Assessment
- Capability
- Communication Network
- Constraint
- Contract
- Cost Type
- Customer Gain
- Customer Job
- Customer Pain
- Customer Relationship Type
- Data Object

Properties @ Any

- Labnaf language properties
- Custom properties (in template package)
- Other (imported, calculated, uncategorized)

Guid
Name
Alias
Author
Complexity
Notes
Phase
Status
Version
(Root Package)
(Parent Packages)
(Parent Elements)



Custom Report

Report Name

Save As Template

Custom Any Report

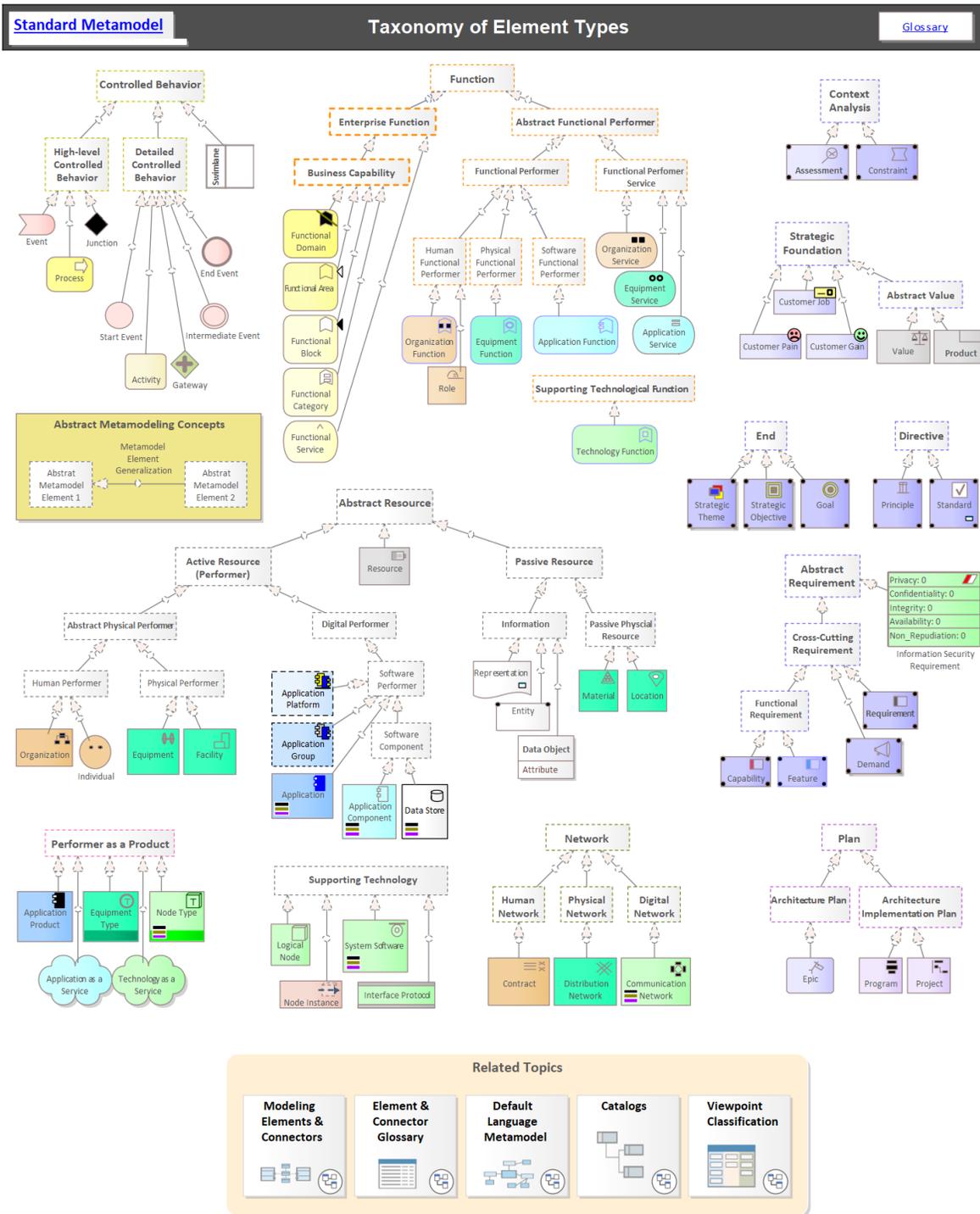
Report Properties

Guid
Name



Language

Many updated Labnaf icons and shapes, making them more readable, optimizing and normalizing text alignment and usage of shape space.



Modeling Elements & Connectors



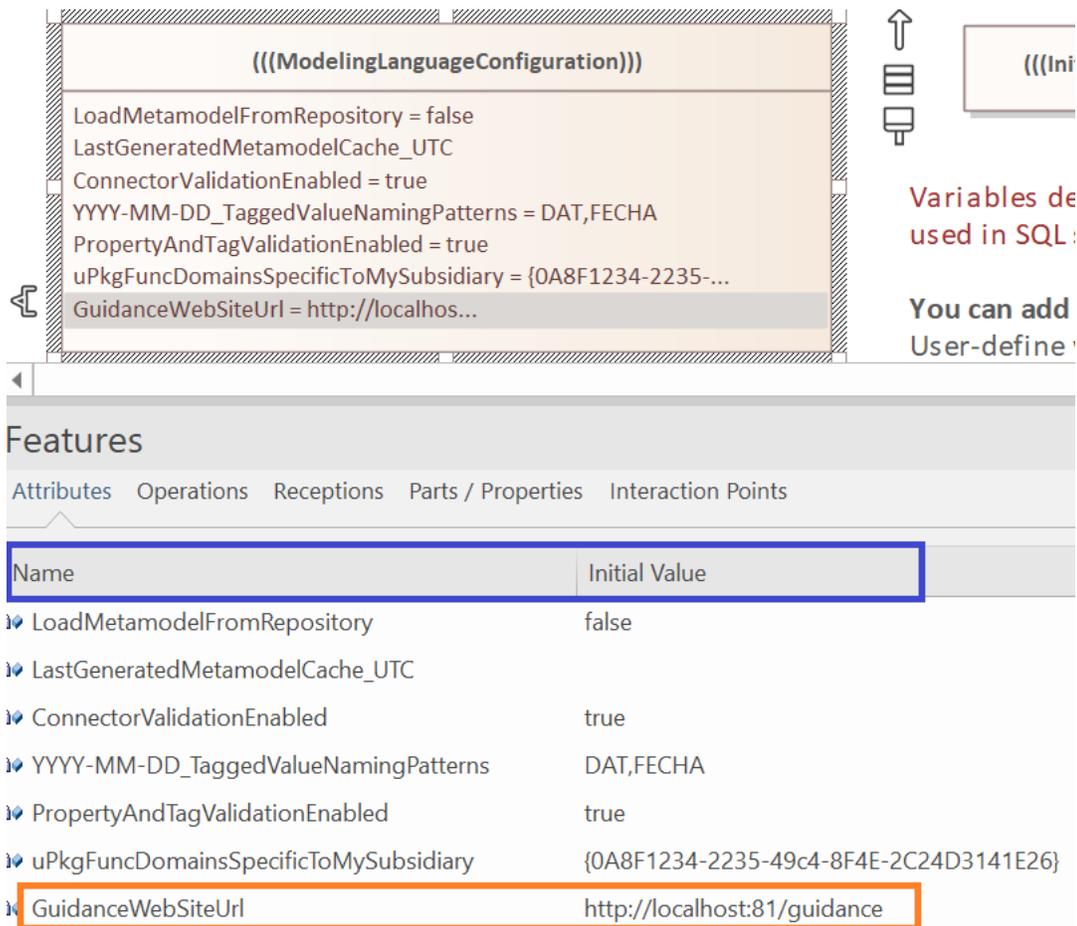
© 2019-2024 Labnaf All Rights Reserved

Labnaf Guidance Help Site Location

The directory structure/file hierarchy of the Labnaf Guidance web site is provided as part of the Labnaf PowerShell package. You can now install a Labnaf guidance website wherever you want including on your own site.

To do so,

- open your repository
- find the core configuration element.
- add an attribute called 'GuidanceWebSiteUrl' to this element and set its default value to your guidance web site url, for example '<http://localhost:81/guidance>'.



The image shows a screenshot of a modeling tool interface. The top part displays a configuration element titled '(((ModelingLanguageConfiguration)))' with several attributes: 'LoadMetamodelFromRepository = false', 'LastGeneratedMetamodelCache.UTC', 'ConnectorValidationEnabled = true', 'YYYY-MM-DD_TaggedValueNamingPatterns = DAT,FECHA', 'PropertyAndTagValidationEnabled = true', 'uPkgFuncDomainsSpecificToMySubsidiary = {0A8F1234-2235-...', and 'GuidanceWebSiteUrl = http://localhos...'. To the right, there are icons for navigation and a box labeled '(((Ini...))'. Below the configuration, there is a section titled 'Features' with tabs for 'Attributes', 'Operations', 'Receptions', 'Parts / Properties', and 'Interaction Points'. A table lists the features with their names and initial values. The 'GuidanceWebSiteUrl' feature is highlighted with an orange border.

| Name | Initial Value |
|---------------------------------------|--|
| LoadMetamodelFromRepository | false |
| LastGeneratedMetamodelCache.UTC | |
| ConnectorValidationEnabled | true |
| YYYY-MM-DD_TaggedValueNamingPatterns | DAT,FECHA |
| PropertyAndTagValidationEnabled | true |
| uPkgFuncDomainsSpecificToMySubsidiary | {0A8F1234-2235-49c4-8F4E-2C24D3141E26} |
| GuidanceWebSiteUrl | http://localhost:81/guidance |

Bug Fix

When the metamodel is customized and the specification of the verbs used in quick linkers is not found, then the specifications of those verbs is generated at the right location [as described here in the documentation](#).

B0608

Language Transformations from the User Interface (Labnaf, ArchiMate, BPMN, UML, SysML...)

This set of interactive features automates the transformation of existing modeling language elements, connectors, and diagrams within a repository by modifying their types and stereotypes.

These features also facilitate the conversion of items from one language to another such as Labnaf, ArchiMate, BPMN, SysML or UML.

This includes

- selecting a range of elements, connectors or diagrams
- changing their type and/or stereotype

Depending on whether you selected an element, a connector or a diagram, the option **Specialize > Labnaf > Change Type** will lead to the appropriate Change Type window.

Each Change Type window offers tooltips that describe the available options.

Access to Change Type features requires specific [user permissions](#).

Only repository administrators can change types across the entire repository or inside the configuration package.

Change Element Type

Change Element Type

Current Element Type

Application Component

Current Detailed Type

Component/LABN::LABN_ApplicationComponent

Use Type of Selected Element

Swap

New Element Type

Application

New Detailed Type

Component/LABN::LABN_Application

Use Type of Selected Element

Generate List Of Element Types

Scope: Where to search for elements with type matching Current Type?

Selected elements or packages Full Package Hierarchy

Elements on active diagram

For Administrators only

All elements in repository Include Configuration Elements

Preview Changes Ask confirmation

Change Type

Help

Close

Change Connector Type

Change Connector Type

Current Connector Type: Component Flow

New Connector Type: Application Flow

Current Detailed Type: InformationFlow/LABN::LABN_ComponentFlow

New Detailed Type: InformationFlow/LABN::LABN_ApplicationFlow

Additional Connector Selection Constraints:

Source Element Type: Application

Target Element Type: Application

Scope: Where to search for connectors with type matching Current Type?

Selected connector, connected elements, or packages with connected elements

Reverse Connectors Direction

Full Package Hierarchy

Connectors on active diagram

For Administrators only

All connectors in repository

Include Configuration Connectors

Buttons: Swap, Clear, Use Type of Selected Connector, Use Type of Selected Element, Preview Changes, Change Type, Help, Close, Generate List Of Connector Types

Change Diagram Type

Change Diagram Type

Current Diagram Type: Application::Application Interaction Details

New Diagram Type: Application::Application Interactions

Current Detailed Type: Object/Labnaf - Application::Application Interaction Details

New Detailed Type: Object/Labnaf - Application::Application Interactions

Additional Diagram Selection Constraints:

Parent Element Type: Application

Scope: Where to search for Diagrams with type matching Current Type?

Selected diagram

Selected package

Full Package Hierarchy

For Administrators only

All diagrams in repository

Include Configuration Diagrams

Buttons: Swap, Use Type of Selected Diagram, Preview Changes, Change Type, Help, Close, Generate List Of Diagram Types

See also the [on-line documentation](#).

Self-documented Metamodels with Abstractions

Labnaf metamodels have always been self-documenting as they are expressed using the same language as the one used for modeling your strategy and architecture.

The ability to create abstract metamodel elements and metamodel element generalizations dramatically diminishes the number of metamodel connections.

And it simplifies the process of creating, modifying, reading, and documenting metamodels.

How to Create a Self-Documented Metamodel with Abstractions

Step1: Create a taxonomy of metamodel elements

This very basic example uses one abstract metamodel element, three concrete elements, and three metamodel element generalization connectors.

Step 2: Define concrete connectors in the metamodel

You can now add concrete elements between abstract and/or concrete elements.

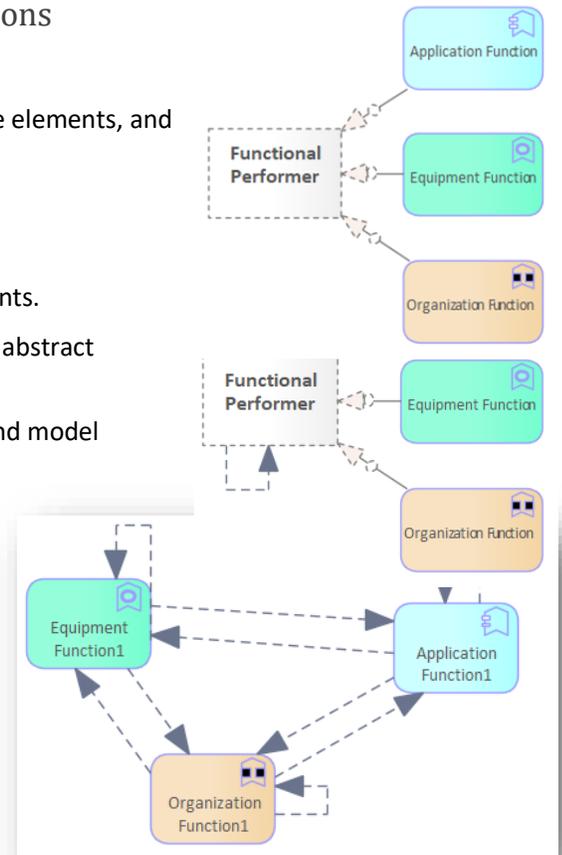
In this basic example, we just added a single concrete 'Flow' connector on the abstract element 'Functional Performer'.

Based on the above metamodel, Labnaf dynamically generates quick linkers and model validation rules.

Step 3 (Usage): Create a model that is compliant with the metamodel

All the model connections shown here are compliant with the above metamodel.

Quick linkers and model validation rules are dynamically generated.



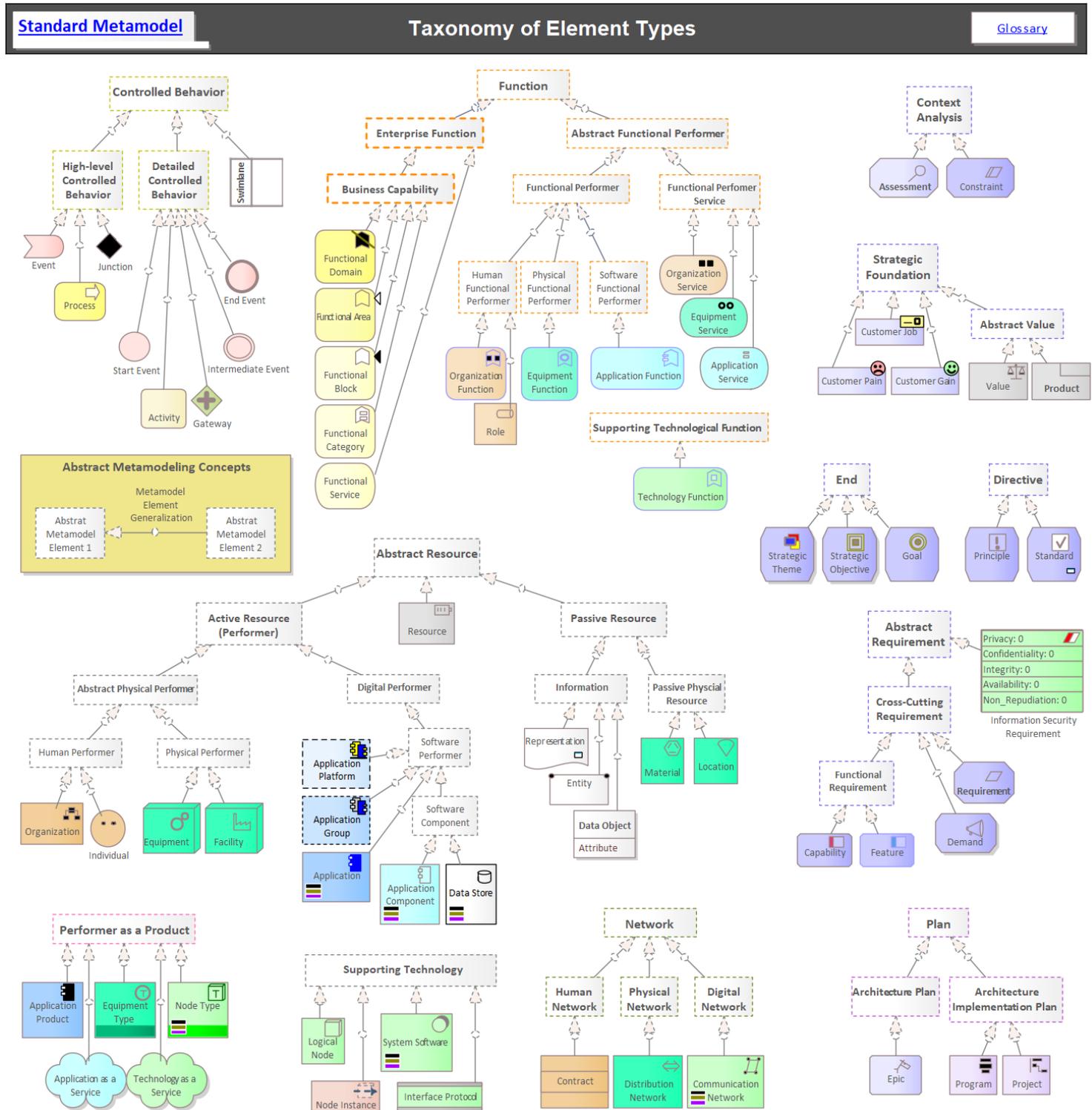
Tool Support

To change a metamodel, you simply add or delete elements and connectors, and then you press the 'Commit Last Changes' button in the Instant Metamodel Manager to commit your changes. The new validation rules become instantly active, and the quick linkers get automatically updated as well.

The Standard Metamodel Uses Abstractions

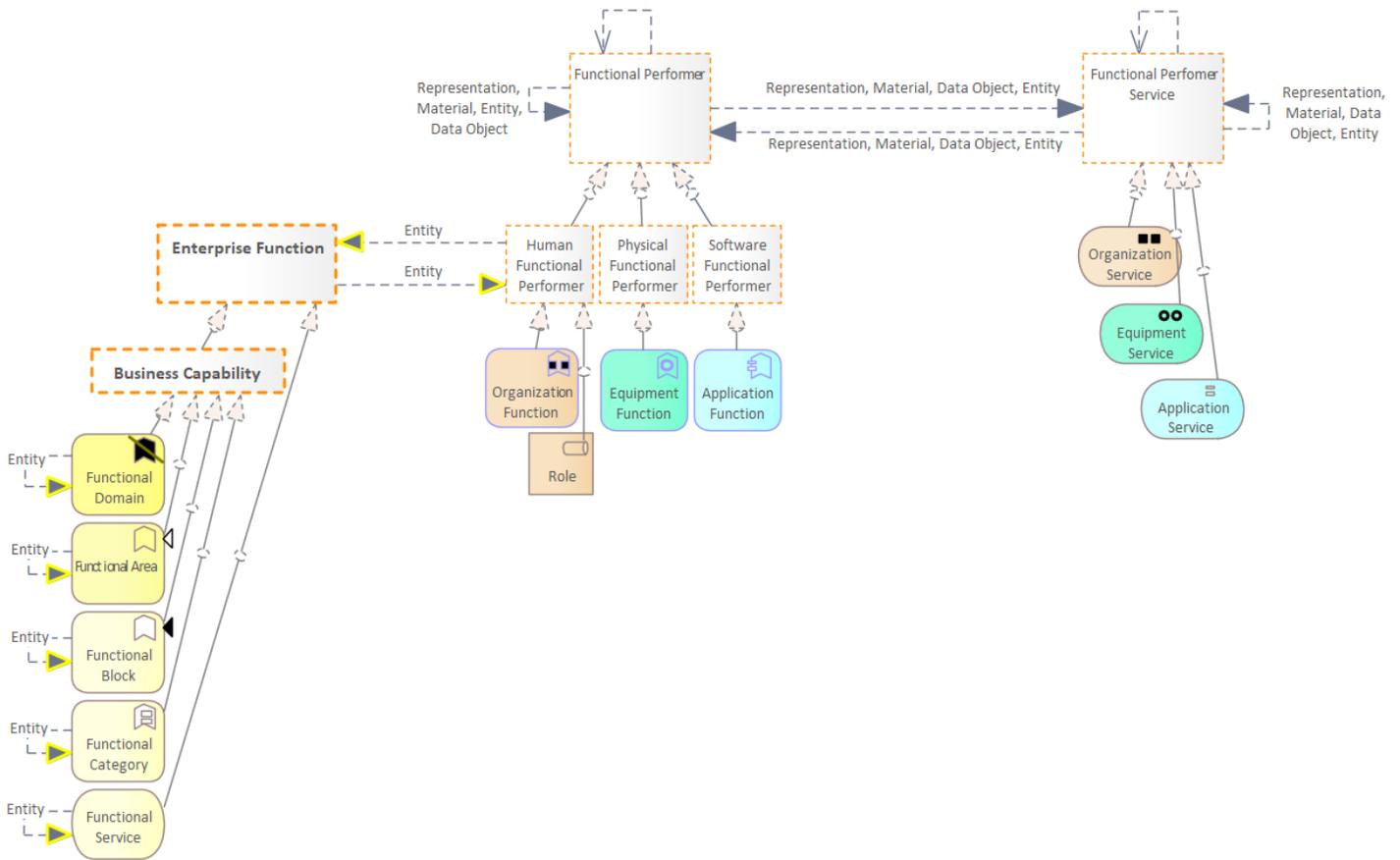
The standard Labnaf metamodel is entirely based on abstract metamodel elements and metamodel element generalizations.

Taxonomy of Element Types

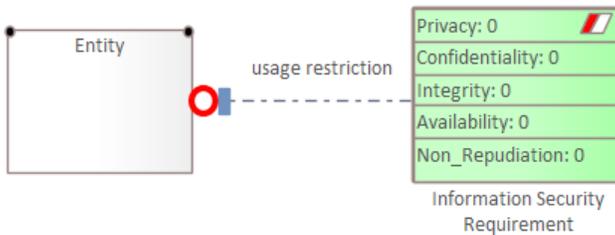


Sample concrete connectors in the standard metamodel

In the following example, most concrete connectors connect abstract metamodel elements.



In the following example, a concrete connector connects concrete metamodel elements.



Upgrading the standard metamodel to a new version of Labnaf

If you upgrade your customized standard metamodel using the [Instant Metamodel Manager](#), Labnaf will correctly integrate the new connectors that you have added in that metamodel with the latest Labnaf changes in that same metamodel.

Visualizing the Standard Metamodel Customizations

Labnaf generates documentation diagrams showing, notably, the connections that you added to the standard metamodel. In other words, these are the differences between the built-in standard metamodel and the customized standard metamodel.

In case your changes to the standard metamodel involve ...

- **only concrete metamodel elements**, like for example, Application or Process, then the generated diagram shows the impacted metamodel elements along with the connectors that you added.
- **some abstract metamodel elements**, like for example, Digital Performer or Function, then the generated diagram contains a textual list of your added connectors.

Labnaf Language Type

New

- New stereotypes and icons for Framework package and for Architecture Management packages



Change

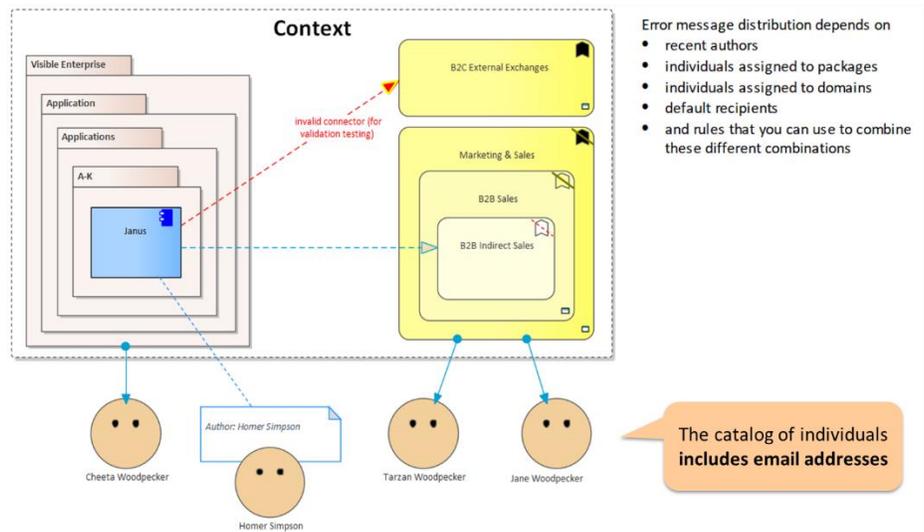
- LABN_Composition stereotype now uses metaclass "Aggregation" instead of "Composition". This is a normalization since Sparx EA now automatically transforms Composition into Aggregation. This should not have any impact on your existing models.

This change was necessary to implement the new Change Connector Type feature.

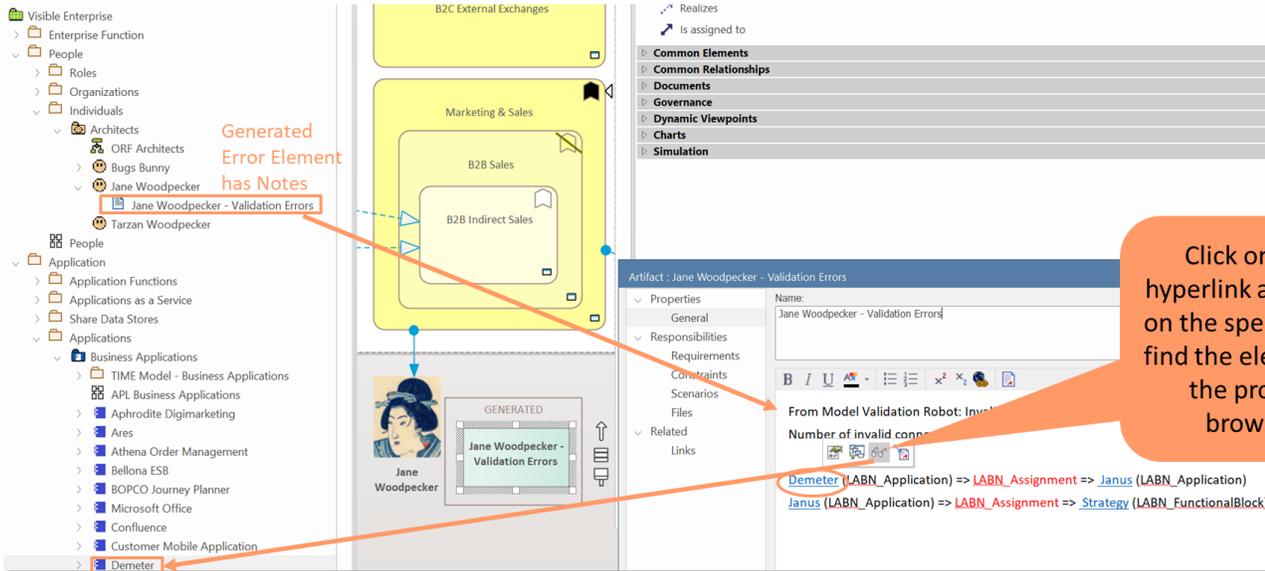
Labnaf PowerShell Validation

Validation errors are now delivered directly in the model repository.

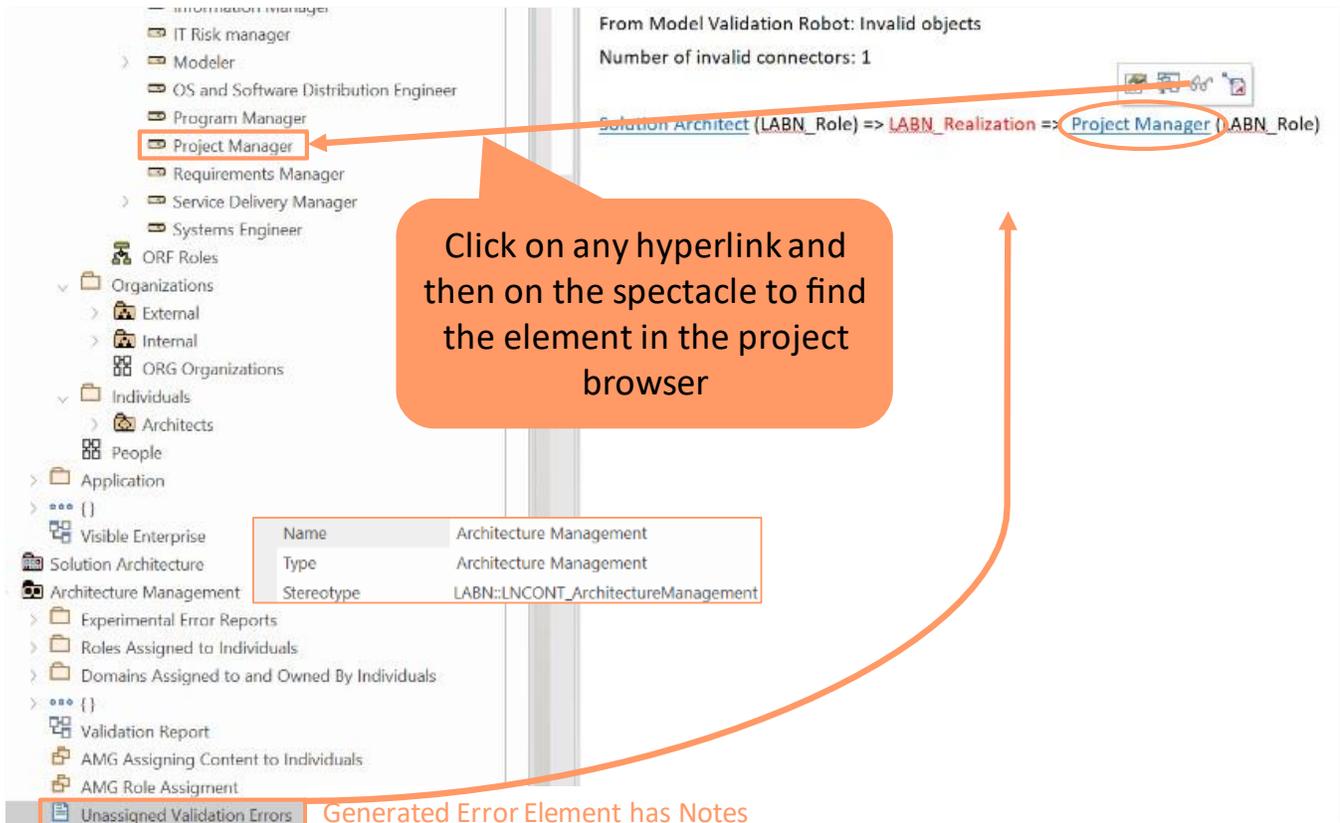
Error message routing is based on architecture management assignments



Error messages are generated as an error element with notes under each assigned individual in the model repository



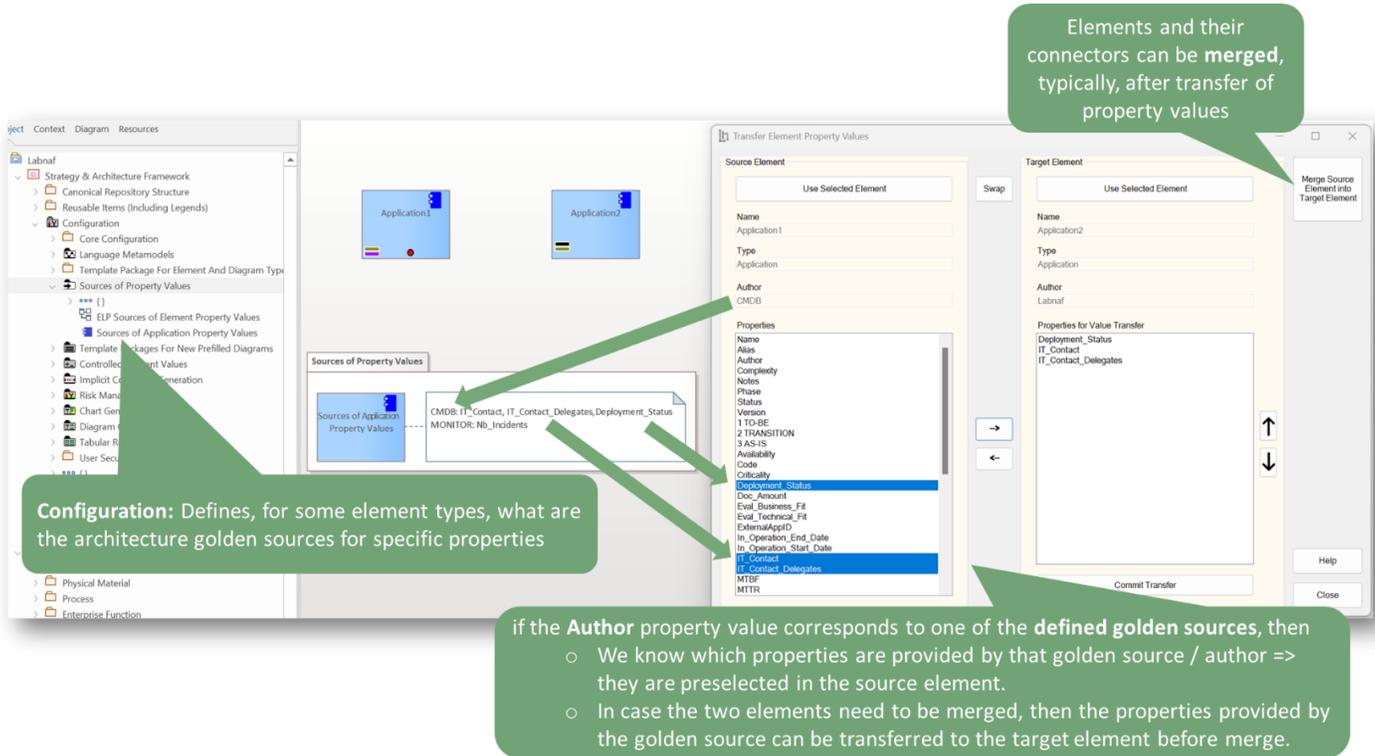
If the error cannot be assigned to any specific individual, then the error element is generated under the unique package of type “Architecture Management”



Transfer Element Property Values

The **'Transfer Element Property Values'** window is used for transferring property values from a source element to a target element. The source and target elements can be set by selecting an element in the browser window or in a diagram, and then by pressing one of the **Use Selected Element** buttons.

When the mouse hovers over a property, then its value is displayed.



The **Left arrow** adds the selected source properties to the target element.

The **Right arrow** removes the selected to-be-transferred properties.

The **Commit Transfer** button effectively transfers properties from the source elements to the target element.

The **Merge Source Element into Target Element** button merges the source element and its connectors into the target element and its connectors. This is typically done after transferring some properties from the source to the target element. The effect is exactly the same as when using the **Merge Versions of Elements & Connectors** feature.

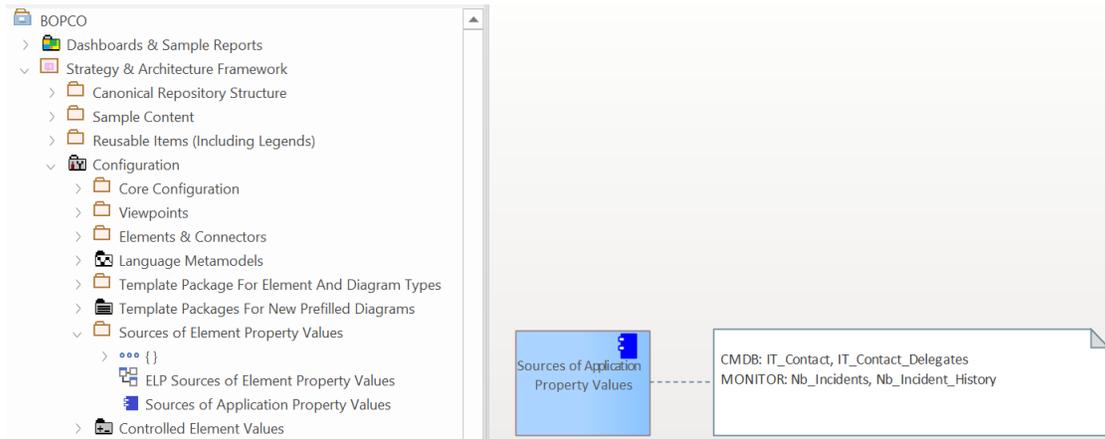
The window is non-modal. Therefore, it allows you to transfer and/or merge from several sources to the same target, or transfer from the same source to several targets.

Of course, you can also achieve the same result by exporting the source element to Excel, changing some properties and then importing the Excel back into the target element. But this is more work.

Sources of Element Property Values

This configuration defines, for some element types, what are the architecture golden sources for specific properties.

The mapping table is stored in the note of a prototype element. Element prototypes, like the one called 'Sources of Application Property Values', are used to define which type of element each configuration applies to.



Effect when Transferring Properties & Merging Elements: if the Author property value corresponds to one of the defined golden sources, then

- We know which properties are provided by that golden source / author => they are preselected in the source element.
- So that, if the two elements are subsequently merged, then the properties provided by the golden source can be preserved.

Instant Metadata Manager

- When a property is renamed, the calculations and source of element property values definitions are automatically updated.
- New property names cannot be SQL SELECT or mathematical expression reserved keywords, as it would jeopardize secure replacement of property names in user-defined SQL statements or mathematical calculations.

Labnaf PowerShell & LanguageTransformer

ExitOnError (new command features appear in green)

Description: When an error is encountered, exit the current script, and optionally send an error email.

Either all email parameters or no email parameter need to be provided.

This command is applicable to Labnaf PowerScripts only.

Usage : LNPS ExitOnError [arguments]

Arguments:

[ErrorMessageSender={ErrorMessageSender}]: On error, a message will be sent from this sender email.

[ErrorMessageSendTo={ErrorMessageSendTo}]: On error, a message will be sent to these email recipients.

[SmtpServerConfigFile={SmtpServerConfigFile}]: Path name of the SMTP Server configuration file.

For further information about the content of the SMTP Server configuration file, please read the section called "SMTP Server Configuration File" in the present document.

Sparx EA Process and Memory Management

- Pressing CTRL-C to interrupt a running script will make sure the related EA process will be terminated as well.
- Workaround Sparx EA memory management issues at the end of command line sessions (e.g. Labnaf PowerShell) to avoid EA memory error message and ghost EA processes in memory on some special situations.

Labnaf PowerShell

Validate (new command features appear in green)

Description: Validate model repository.

Usage: LNPS Validate [arguments]

Arguments:

RepoPathName: Repository path name (EAP file). ** NOT NEEDED IN A LABNAF SCRIPT FILE (.lpssc) **

ValidationConfigurationFile: Path name of the model validation configuration file.

SmtpServerConfigurationFile: Path name of the SMTP Server configuration file.

[IgnoreTemporaryTraceConnections] : Indicates that Temporary Trace Connections will not be reported as errors.

Bug fixed

- Generate Tabular Report form no longer truncated on small (laptop) screen)
- Log files created at the exact same second are renamed so that they are no longer mixed up

Labnaf AddIn Lite Edition (free; no end date; unlimited number of elements)

No more limitation on the number of Labnaf elements that can be created.

Full support for local repository files (SQLite / QEA).

SQL server repositories are supported for browsing Labnaf repository content.

Labnaf PowerShell Database Commands

DeleteImportedConnections (new command)

Description: Delete imported connections.

To prevent any error from causing damage, this command can only delete imported connections.

Usage : LNPS DeleteImportedConnections [arguments]

Arguments:

- **RepoPathName:** Repository path name (EAP file). **** NOT NEEDED IN A LABNAF POWERSCRIPT FILE (.lpsc) ****
- **ALL | Unlabeled | Label={Label} | SqlFileName={SqlFileName}:** Defines which imported connections will be deleted.
 - **All:** Delete all (labeled or unlabeled) imported connections.
 - **Unlabeled:** Delete unlabeled imported connections.
 - **Label={Label}:** Delete imported connections labeled {Label} (see ImportConnections).
 - **SqlFileName={SqlFileName}:** Delete imported connections selected by a query stored in a file.
The query is expressed as a SQL statement selecting the IDs of the connections to be deleted.

ExitOnError (new command)

Description: Exit the current script when an error is encountered. This command is applicable to Labnaf PowerScripts only.

Arguments

- **** NONE **:** This command does not require any argument.

ImportConnections (new command features appear in green)

Description: Import connections from a CSV or Excel file.

Identify the 'From' and 'To' elements to be connected using

- EITHER an ea_guid,
- OR an optional element stereotype + a unique key that can be a property or tagged value. To define a unique key, you simply add a '#' in front of the property or tag name
 - If multiple unique keys are provided, they are searched in this order: Tagged Value, Name, Alias

Create the connection if unique elements are found for the two sides of the connection.

In the input file, the first line can contain the following headers:

```
ConnectorType,  
ConnectorStereotype,  
From#Type  
From#Stereotype,  
From#GUID,
```

From#Name,
From#MyUniqueKey,
From#Alias,
To#Type,
To#Stereotype,
To#GUID,
To#Name,
To#MyUniqueKey,
To#Alias

Providing either the ConnectorType or the ConnectorStereotype is mandatory so that the connectors can be created. Same for the From and Target element types.

The stereotype suffices for languages known by the Labnaf PowerShell, as it will derive the type from the stereotype.

If a field name mapping file (CSV) is provided, the first line must contain the following headers:

Input_Column_Names, Target_Column_Names

Usage: LNPS ImportConnections [arguments]

Arguments:

- **RepoPathName:** Repository path name (EAP file). **** NOT NEEDED IN A LABNAF SCRIPT FILE (.lpsc) ****
- **SourceFile:** A CSV or Excel file containing the data that needs to be imported.
The name of the CSV file can define the CSV character encoding and column delimiter.
The syntax is: '...[CP={code page number or name};DELIM={COMMA|SEMICOLON|TAB|SPACE|a character code number}]...CSV'
If no format is specified in the file name, then the default values are '[CP=utf-8;DELIM=COMMA]'
- For further information, see ['Inbound/Outbound Content Formatting'](#) on the Guidance Web Site
- **[ColumnMappingFile]:** An optional CSV file containing the mapping between the input and output column names
- **[SourcePackageGuid={SourcePackageGuid}]:** The GUID of the package where the connection source elements are located
- **[TargetPackageGuid={TargetPackageGuid}]:** The GUID of the package where the connection target elements are located
- **[SourcePackageStereotype={SourcePackageStereotype}]:** The stereotype of the package where the connection source elements are located
- **[TargetPackageStereotype={TargetPackageStereotype}]:** The stereotype of the package where the connection target elements are located
- **[SetConnectorLabel={LabelValue}]:** A label that will be added to the alias of the imported connections.
=> All imported connections will have an alias with the value '@[{LabelValue}]'

Generate Tabular Report

- Supports all standard properties: Alias, Author, Complexity, Name, Notes, Phase, Priority, Status, Version.

Import (AddIn or PowerShell)

- Supports all standard properties: Alias, Author, Complexity, Name, Notes, Phase, Priority, Status, Version.
- Improved performance

Calculations

Arithmetic Calculation (AddIn or PowerShell)

- Empty property values involved in calculations are replaced by 0

New Value Control Behavior



If you want some periodical calculation template to be ignored by the Labnaf PowerShell, then add the prefix "--" to its name.

You will still be able to run it from the modeling environment.

If you build calculations that work only with SQL Server then include "SQLSVR" anywhere in the value control name.

They will be ignored in the case where the database is not SQL Server. The Labnaf PowerShell will issue a warning.

Labnaf PowerShell Database Connections

- Support both native and OLEDB connections to SQL repositories.

SQLite / QEA Repository File Support

Almost all commands now support SQLite / QEA file-based repositories.

Remaining limitation: Implicit connection generation is not supported (too complex for SQLite database capabilities).

Language Transformer

- All commands are now compliant with SQLite/QEA repositories.
- ChangeConnectorType: Sequence of Arguments changed to comply with Labnaf PowerScript general syntax.

Bug fix

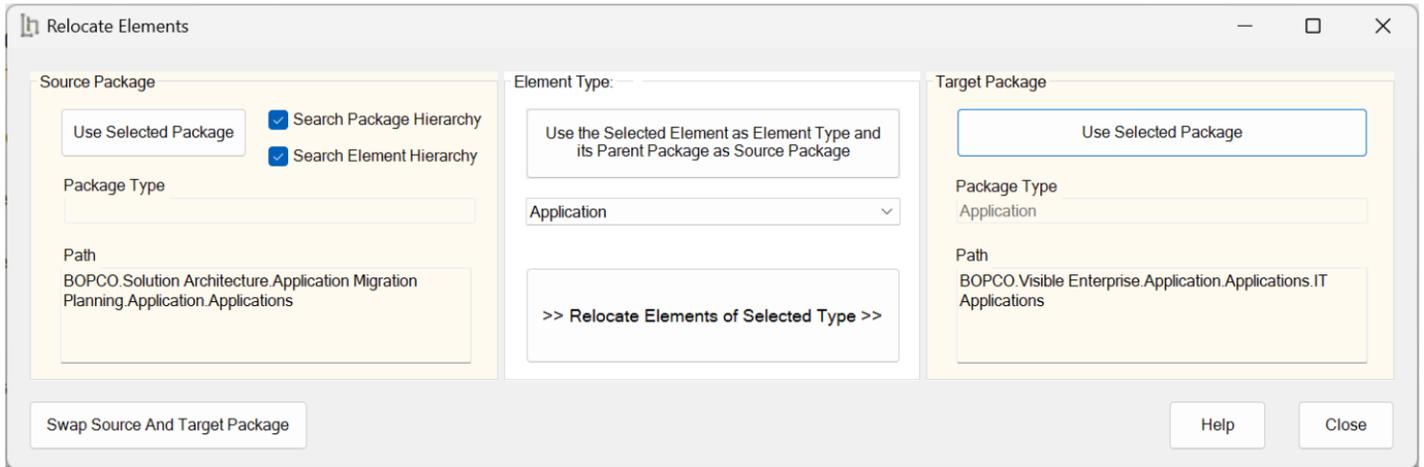
The Generate Tabular Report function now properly reports Notes and Memo fields.

Relocate/Classify Elements

Using the "Relocate Elements" window, you can quickly Relocate all elements of a selected type from a source package to a target package.

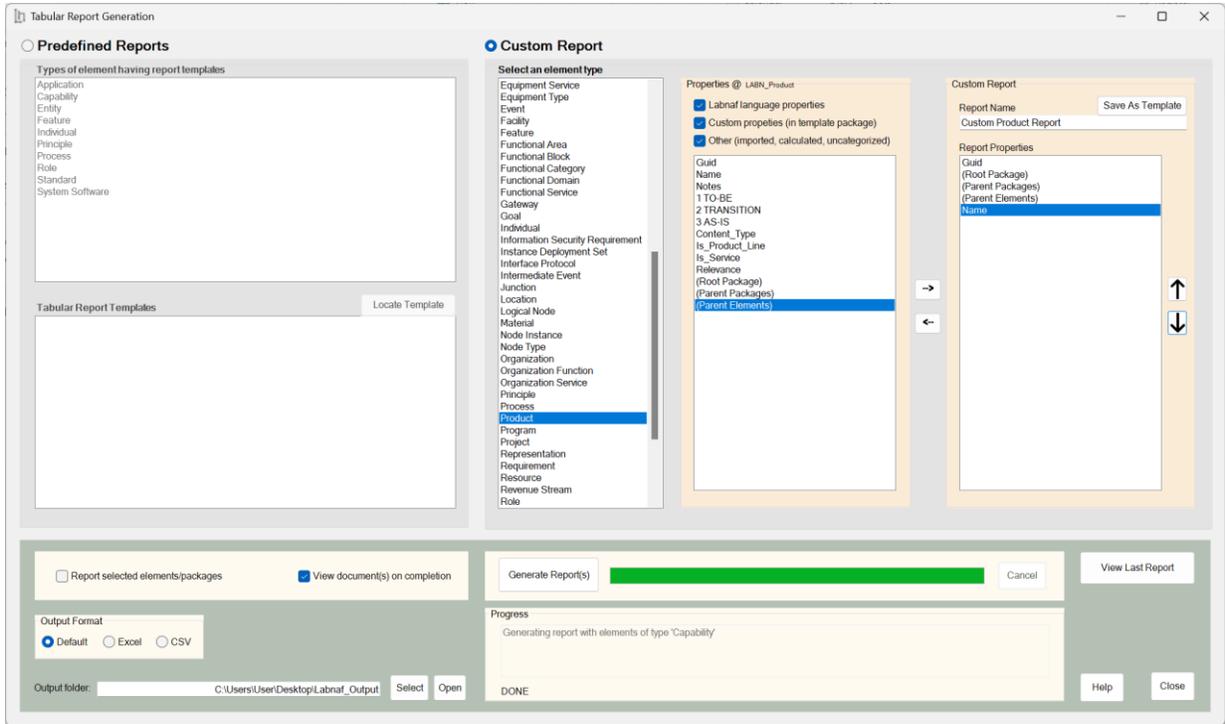
This is typically used for moving scattered elements to the catalog package that they belong to.

Select the source package, the target package, and the type of element to be moved from the browser window or from diagrams. Use the checkboxes to indicate that the elements that you wish to move may be located in sub-packages and/or in sub-elements.



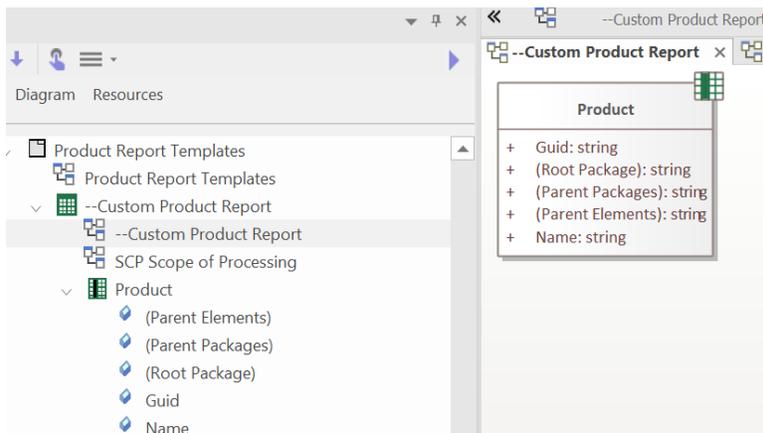
Generate Tabular Report

Reports can now include special columns: (Root- Package), (Parent Packages), and (Parent Elements).



| | Product: Guid | (Root Package) | (Parent Packages) | (Parent Elements) | Name |
|----|---------------|------------------------|---|---|---|
| 1 | | | | | |
| 2 | 76AA5B | Labnaf (BOPCO Example) | Labnaf (BOPCO Example).Vision.Strategy Definition.Corporate Strategic Foundations.Value Propositions.Products | | BOPCO International Travel Product Line |
| 3 | 6F2F6A | Labnaf (BOPCO Example) | Labnaf (BOPCO Example).Vision.Strategy Definition.Corporate Strategic Foundations.Value Propositions.Products | | BOPCO National Commuter Product Line |
| 4 | 885BDB | Labnaf (BOPCO Example) | Labnaf (BOPCO Example).Visible Enterprise (Enterprise Architecture).Generic.Products | | Digital Product |
| 5 | EEC2D5 | Labnaf (BOPCO Example) | Labnaf (BOPCO Example).Vision.Strategy Definition.Corporate Strategic Foundations.Value Propositions.Products | BOPCO International Travel Product Line | International Ticket |
| 6 | 9CD42A | Labnaf (BOPCO Example) | Labnaf (BOPCO Example).Vision.Strategy Definition.Corporate Strategic Foundations.Value Propositions.Products | BOPCO International Travel Product Line | International Travel Subscription |
| 7 | 56DD32 | Labnaf (BOPCO Example) | Labnaf (BOPCO Example).Vision.Strategy Definition.Corporate Strategic Foundations.Value Propositions.Products | BOPCO National Commuter Product Line | National Travel Subscription |
| 8 | 0799DA | Labnaf (BOPCO Example) | Labnaf (BOPCO Example).Vision.Strategy Definition.Corporate Strategic Foundations.Value Propositions.Products | BOPCO National Commuter Product Line | National Travel Ticket |
| 9 | E06EDC | Labnaf (BOPCO Example) | Labnaf (BOPCO Example).Vision.Strategy Definition.Corporate Strategic Foundations.Value Propositions.Products | BOPCO International Travel Product Line | Onboard Meals |
| 10 | 960EAB | Labnaf (BOPCO Example) | Labnaf (BOPCO Example).Vision.Strategy Definition.Corporate Strategic Foundations.Business Models.Products | | Product A |
| 11 | 3508EE | Labnaf (BOPCO Example) | Labnaf (BOPCO Example).Vision.Strategy Definition.Corporate Strategic Foundations.Business Models.Products | | Product B |

Such a report can also be "Saved as template". The template can then be referenced by (scheduled) reporting scripts.



Search

New built-in search selecting 3 levels of enterprise functions / business capabilities and leaving redundant parent names as blank.

Search name: "Catalog – Functional Blocks – Blank Same Values"

Sample result:

| Functional_Domain | Functional_Area | Functional_Block | Value_Type | Efficiency |
|-------------------------|-------------------------------------|--|------------|------------|
| B2B External Exchanges | Financial Exchanges | Payment Exchanges | Financial | 3 |
| B2C External Exchanges | Digital Communication | Email Exchange | Other | 3 |
| | | SMS Exchange | Other | 3 |
| Finance | Accounts Payable | Labor Funding | ? | ? |
| | Customer Payment | Card Payment | Financial | 3 |
| | | Cash Payment | Financial | 4 |
| | | Payment Confirmation | Financial | 4 |
| HR & Corporate services | Compensations & Benefits Management | Benefits Management | ? | ? |
| | | Compensation Management | ? | ? |
| | | Terms Management | ? | ? |
| | Facilities Management | Space Allocation | ? | ? |
| | Recruitment and Hiring | Employee Supply and Demand Management | ? | ? |
| | | Onboard Tracking | ? | ? |
| | | Position Advertizing | ? | ? |
| | | Skills Assessment | ? | ? |
| Information Management | Artificial Intelligence | Deep Learning Model | Other | 4 |
| | | Expert Systems | Other | 4 |
| | | Fuzzy Logic Model | Other | 4 |
| | | Knowledge-Based Systems | Other | 4 |
| | | Machine Learning Model | Other | 4 |
| | | Natural Language Processing | Other | 4 |
| | | Neural Network Model | Other | 4 |
| | | Robotic Process Automation | Other | 5 |
| Information Technology | Detect to Correct (D2C) | Change Control | Other | 3 |
| | | Configuration Management | Other | 3 |
| | | Release Composition (Change & Release Management | Other | 3 |
| | | Service Level | Other | 3 |
| | Requirement to Deploy (R2D) | Service Design | Other | 5 |
| | Strategy to Portfolio (S2P) | Enterprise Architecture | Other | 5 |
| | | Policy | Other | 5 |
| | | Portfolio Demand | Other | 5 |
| | | Proposal | Other | 5 |

Import Tabular Report (improvements in UI and PowerShell)

- Memo properties are now properly imported, including formatting.
- CSV import problem introduced in last release has been fixed.

Labnaf PowerShell

UpdateAllDiagramImages (new command)

Description: Update all diagram images in a model repository.
Diagram images are used by Prolaborate when it presents diagrams in its user interface. These images are created when you manually save a diagram in Sparx Systems' EA. But they are not updated when you change an element or connector (for example rename) by other means.

Usage: "C:\Program Files\Labnaf\PowerShell\Inps.exe" UpdateAllDiagramImages [arguments]

Arguments:

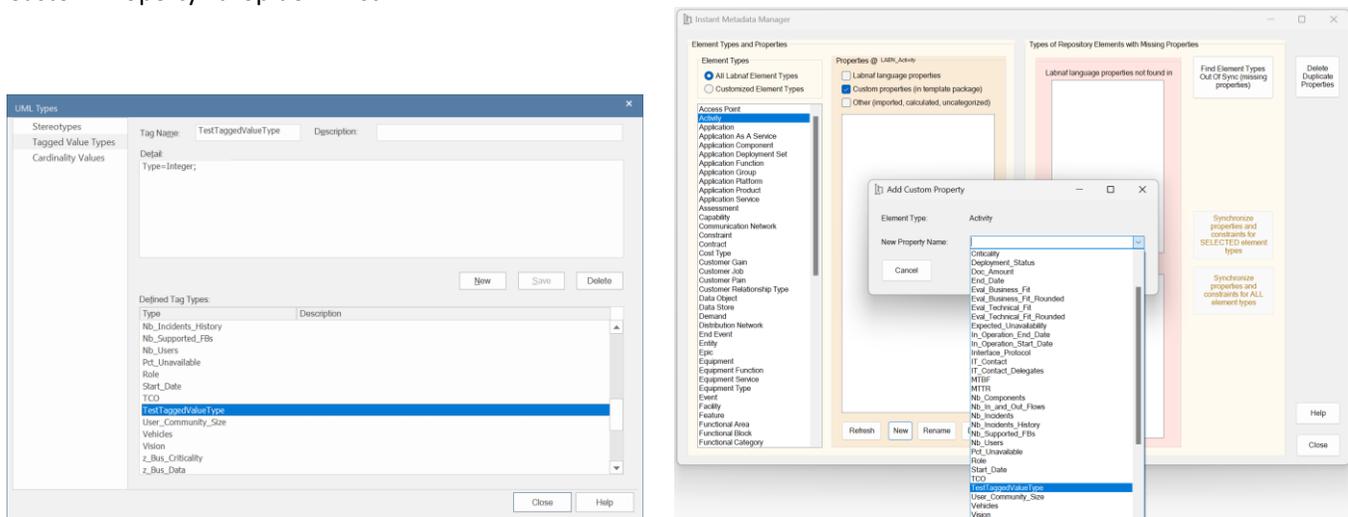
- RepoPathName: Repository path name (EAP file). ** NOT NEEDED IN A LABNAF POWERSCRIPT FILE (.lpsc) **

Implicit Data Generation (improved)

- All expected implicit aggregations are now generated for embedded elements.

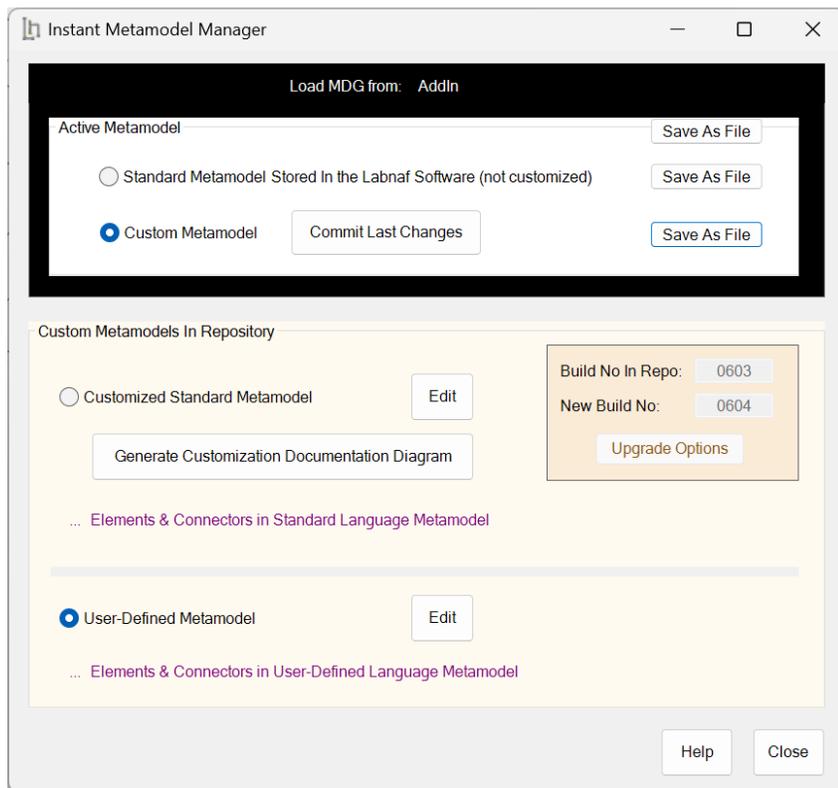
Instant Metadata Manager

When a tagged value/property type is added or deleted, this immediately impacts the list of properties available in the “Add Custom Property” drop down list.



Instant Metamodel Manager

New buttons to save metamodels as files. The files can then be compared using free tools like, for example, WinMerge. This is an alternative to generating a Customization Documentation diagram.



In Build 0604, other users will no longer get a message saying that the metamodel has changed.

Advanced Application Value Management

Application TIME Models

Legacy applications frequently demand more attention than what the budget allows. The Gartner TIME framework (Tolerate, Invest, Migrate, Eliminate) stands as a renowned industry method for strategizing and fine-tuning application portfolio enhancements.

This strategy provides a comprehensive avenue to optimize both the business and technological aspects of an application portfolio, ensuring each application aligns well with the organization's requirements.

To facilitate this, application leaders often resort to the TIME categorization, visualized as a bubble chart for their applications. Clicking on these bubbles reveals in-depth details about the respective applications. This visualization aids in prioritizing portfolio improvements by considering factors such as business and technological suitability, associated risks, and costs.

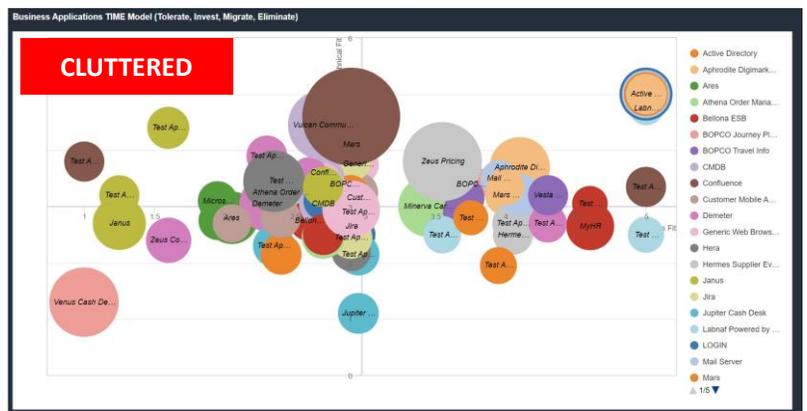
Labnaf's approach delves deeper by evaluating the business and technological fit grounded on specific criteria. This includes, but is not limited to, aspects like business satisfaction, potential, technical maintainability, and scalability. The flexibility of this system allows for criteria to be seamlessly added, deleted, renamed, or weighted as needed, with the resulting fits being recalculated in real-time.

Uncluttering TIME models

As the quantity of applications in our portfolio expands, the clarity of the TIME models becomes compromised, rendering them less functional and more chaotic. In the TIME model depicted at the bottom-right, each bubble symbolizes an individual application. Even with just a few scores of applications displayed on the second chart, it's evident how readability is quickly challenged due to overlapping bubbles.

Conversely, in the top-right TIME model, every bubble stands for a cluster of applications. By clicking on any given bubble, you can delve deeper into that group. To maintain clarity, Labnaf organizes applications with analogous business and technological alignments into specific groups. This level of organization can be readily adjusted to ensure optimal visibility of the TIME models.

Should there be a need for multiple TIME models, perhaps domain-specific, distinct grouping criteria can be designated for each, contingent on the number of applications associated with each domain.



Implementation Overview

The uncluttered TIME model calculation and generation is implemented using a low code Labnaf Power Script that you can run either once, or that you can schedule following your preferences. Labnaf Power Scripts are part of the Labnaf PowerShell environment.

By default, the process runs during the day. It periodically recalculates the business fit and technical fit, refreshes the uncluttered TIME models, and finally stops in the evening.

The detailed application evaluation criteria, like business satisfaction, potential, technical maintainability, and scalability, can be easily and quickly adapted and extended. Same for the calculation of the business fit and technical fit.

Example: Business Fit = Bus_Data*25/100 + Bus_Needs*30/100 + Bus_Potential*15/100 + Bus_Satisfaction*30/100

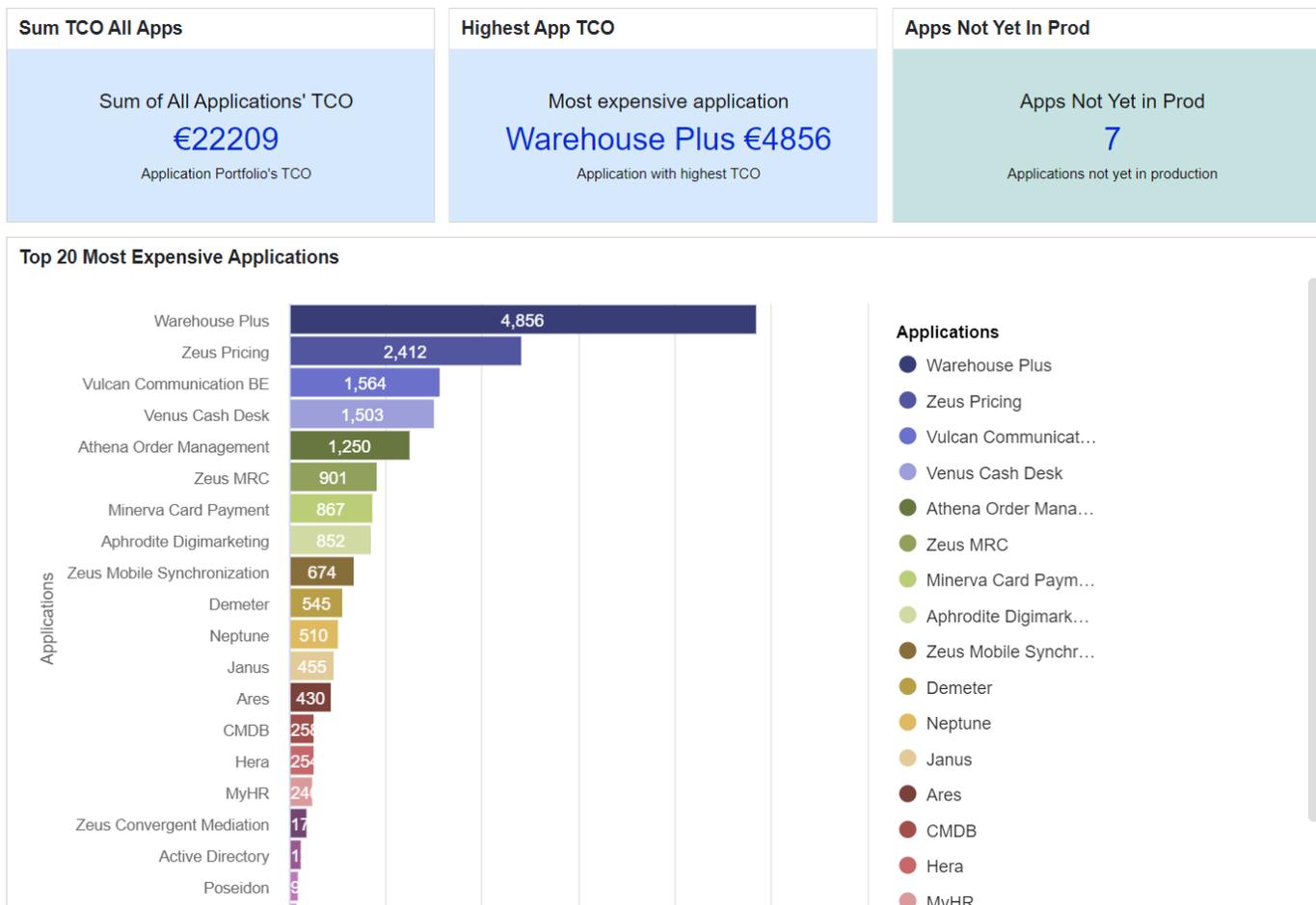
You can generate multiple TIME models, for different application categories, and following different grouping criteria that can be easily configured.

The solution comes with Prolaborate widget specifications: TIME Model bubble chart, landscape chart and report.



| Application... | Business_fit | Technical_fit | Tco | Criticality | Nb_users | Nb_incidents | Vision | It_contact |
|---------------------|--------------|---------------|------|-------------|----------|--------------|-----------|------------|
| Venus Cash Desk | 1 | 1.3 | 1503 | M | 4 | 5 | Phase Out | Doc |
| Janus | 1.25 | 2.7 | 455 | M | 4 | 6 | New | Happy |
| Zeus Convergent Med | 1.6 | 2.4 | 173 | M | 3 | 6 | Maintain | Sleepy |
| Microsoft Office | 1.95 | 3.1 | 49 | M | 5 | 5 | Maintain | Happy |
| Ares | 2.05 | 2.8 | 430 | L | 5 | 5 | Maintain | Grumpy |
| Demeter | 2.3 | 3.05 | 545 | M | 4 | 4 | Maintain | Bashful |
| Zeus MRC | 2.35 | 3.45 | 901 | M | 3 | 5 | Maintain | Bashful |
| Zeus CCE | 2.4 | 2.8 | 56 | M | 1 | 1 | Maintain | Sneezy |
| Athena Order Manage | 2.4 | 3.25 | 1250 | M | 3 | 7 | Invest | Happy |
| Bellona ESB | 2.6 | 2.75 | 69 | M | 3 | 5 | Maintain | Sneezy |

Other Application Value Management charts:



See also the [TIME Model Generation Configuration Guide](#).

Labnaf PowerShell – Important Changes

See also the [Labnaf PowerShell Reference Guide](#)

Labnaf (low code) PowerShell

Using the Labnaf PowerShell, you could already execute commands from a Windows script file (.cmd). You can now also execute commands from a **Labnaf PowerShell file (.lpsc)** as detailed below. Here are some examples...

Run the script only ones

```
"C:\Program Files\Labnaf\PowerShell\Inps.exe" Enrich_And_Distribute_Architecture_Data.lpsc Repository.eap
```

Run and restart the script after 120 seconds; Stop restarting after 720 minutes following the initial start

```
"C:\Program Files\Labnaf\PowerShell\Inps.exe" Enrich_And_Distribute_Architecture_Data.lpsc Repository.eap 120 720
```

Every morning at 8:00:00, run the script and restart 120 seconds after completion; Stop restarting after 720 minutes following the initial start i.e. at 20:00:00

NB: One day = 1440 minutes; 12 hours = 720 minutes.

"C:\Program Files\Labnaf\PowerShell\Inps.exe" ScheduleCommand RunScript.cmd 08:00:00 1440

Content of RunScript.cmd: "...\Inps.exe" Enrich_And_Distribute_Architecture_Data.Ipsc Repository.eap 120 720

Sample Contents of 'Enrich_And_Distribute_Architecture_Data.Ipsc' PowerScript file

```
// Labnaf PowerScript File
/* Can include comments using common programming language delimiters.
   There is no parameter specifying the applicable repository since the repository is open once for the entire script */

ImportTabularReport "..\Input\ImportTabularReport\Import - Application Incident History.xlsx" - LABN_Application

GenerateImplicitData

CalculateTaggedValues

GenerateDiagrams "{8D6CDB63-5FC3-4714-9BA3-440E94E41907}"

BackupToZipFile "%OUTPUT_DIR%\RepositoryBackups" "%OUTPUT_DIR%\Tmp"

Validate "..\Input\Validate\ValidationConfigs.xml" "..\Input\Validate\SmtpServerConfig.xml"

GenerateTabularReports "..\Output"

GenerateDoc "..\Output\GeneratedDocument.docx" "{E07513A3-0C37-42a9-8F3C-AE7AD3A61715}" "Generic Report -
Diagrams Only"

GenerateHTML "..\Output\HTML" "{E07513A3-0C37-42a9-8F3C-AE7AD3A61715}" LABN
```

- **Variables:** Labnaf **PowerScript** files can set and use environment variables. **SQL queries** that are used by the PowerScript commands can also use these variables.

Example:

```
GenerateTabularReports %OUTPUT_DIR% Capability /* OUTPUT_DIR is variable that was set earlier */
GenerateTabularReports %OUTPUT_DIR% LABN_Application "Application List[CP=utf-8].CSV"
```

The **PowerScript** is available for the **Labnaf PowerShell** and for the **Labnaf Language Transformer**.

Command ImportTabularReport (improved)

- The command can now import elements from any modeling language using tis parameter:

ElementTypeAndOrStereotype: The type and/or stereotype of the elements that need to be created or updated.

If both type and stereotype are provided, they must be separated by a '/'.
Example: 'Component/LABN_Application'.

If only a type is provided it must be a valid Sparx Systems' EA element type.

Example: 'Class'.

If only a stereotype is provided it must be a Labnaf element stereotype.

Example: 'LABN_Application'.

If the stereotype is not a Labnaf stereotype, then the type is mandatory, and the stereotype must be fully qualified.

Example: 'Component/ArchiMate3::ArchiMate_ApplicationComponent'

- To define a unique key, you can add '#' in front of the property or tag name in the input file. Alternatively, you can now add 'UNIQUE_KEY_' instead. This is useful in case the report to be imported is itself generated and the '#' character is reserved.

Command BackupToZipFile (New)

Description: Backup a DBMS repository to a date-stamped, zipped .qea file.

Usage: LNPS BackupToFile [arguments]

Arguments:

- **RepoPathName:** Path name of the source database repository (EAP file containing a connection string). ** NOT NEEDED IN A LABNAF POWERSCRIPT FILE (.lpsc) **
- **DestPathName:** Path name of the destination file-based repository (.qea, .qeax, .feap, eadb, .eap, .eapx,).
- **DetailedLogFilePath:** Alternative path for the detailed log file (supersedes the default detailed log file path).

Command DeleteSelectedElements (New)

Description: Delete elements selected by a SQL SELECT command stored in a file. To prevent any error from causing damage, this command can only delete generated or imported elements.

Arguments:

- **RepoPathName:** Repository path name (EAP file). ** NOT NEEDED IN A LABNAF SCRIPT FILE (.lpsc) **
- **SqlSelectElementIdsFileName:** The path name of a file that contains a SQL statement to select the element IDs of the elements to be deleted

Set (New)

Description: Set a Windows environment variable. The variable can be used in the script and in the SQL statements that are referenced by commands used in the script. The variable will be available only to the current process and its child processes.

Usage: LNPS Set [arguments]

Arguments:

- **Variable=String:** Name and value of the new or existing variable.

CmdValidate (changed)

- More detailed log and exception report when sending messages to individuals

Example:

Validate: Start (2023-10-11 21:36:21)

Labnaf commands log dir: C:\A\LT\SparxDev\Test\OUTPUT\PowerShell\SCHEDULED\Log

Loading validation configuration.

Loading smtp server configuration.

Loading potential error message recipients.

Found 15 individuals.
Loading validation rule set.
- Loading MDG from Labnaf AddIn
- Loading connection validation rules from the standard metamodel built in Labnaf
Running validation rule set.
Number of invalid connectors found: 1
Assigning error messages to individuals.
Sending error messages to individuals.
Sending error message
From: model.validation@labnafdemo.com
To: labnaf@labnaf.local
SMTP Server Configuration
Smtp Delivery Method: Network
Host: 127.0.0.1
Client Port: 25
Enable SSL: False
User Name: alain@labnafdemo.com
Password: [See conguration]
ERROR - Could not send message. Detailed Error:
Failure sending mail.
Unable to connect to the remote server
No connection could be made because the target machine actively refused it 127.0.0.1:25
Validate: Done (2023-10-11 21:39:23) - Elapsed time: 00:03:01.8107688

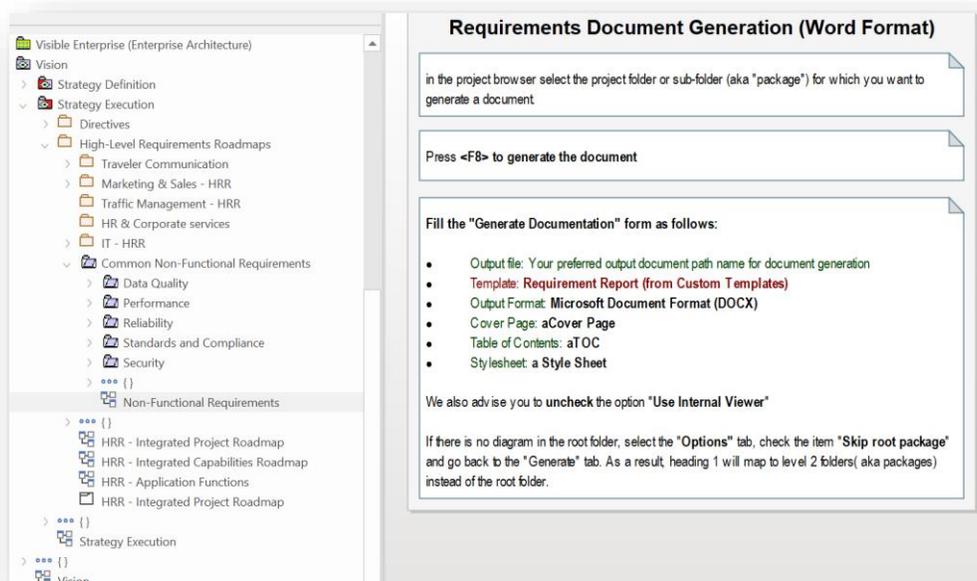
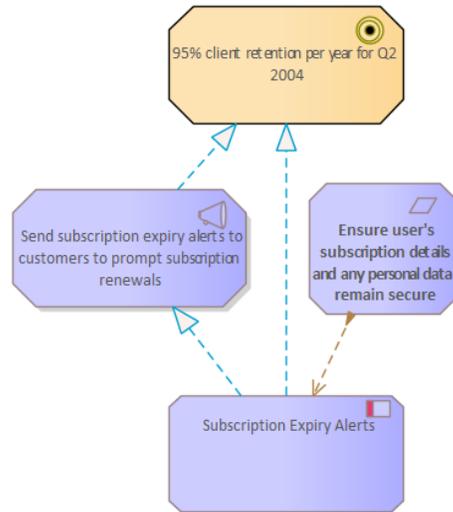
Tabular Report Generation

- HTML text stored in element Notes and Memo properties/tagged values is now exported to plain text in Excel and CSV files.
- Ability to create reports on elements and sub-elements of a selected type stored in packages that are selected either in the project browser or in the active diagram.

B0602

Language

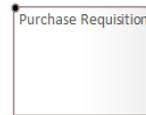
- Default properties removed and [provided as loadable configuration files \(Tagged Value Types and Template Application\)](#) that can be also added, renamed or deleted using the Instant Metadata Manager.
 - For Application
 - All tagged values except 1 TO-BE, 2 TRANSITION, and 3 AS-IS
 - For Functional Domain, Functional Area and Function Block
 - Nb_Incidents, TC_Supporting_Apps, Avg_Nb_Supporting_Apps
- New Element Type: Requirement with corresponding type of catalog package.



Excerpt from the Labnaf [Glossary](#):

| | |
|--|--|
| | <p>A demand is a request to change some aspects of the enterprise landscape. A demand can be strategic or operational.</p> <ul style="list-style-type: none"> • Strategic demands realize corporate or domain-specific goals. • Operational demands do not realize any strategic goal whatsoever. An operational demand might arise for example to resolve some operational continuity issue.. |
| | <p>A requirement element represents a condition or characteristic that an architecture description must satisfy to meet some goals or principles. A requirement can be functional or non-functional. A requirement can influence some target capabilities which are used for solution planning and roadmapping purpose.</p> |
| | <p>A (target) capability is a required high-level system behavior that an architecture description must satisfy to meet goals. (Target) capabilities are delivered by (architecture) epics. The envisioned system behavior can be performed by software, people and equipment.</p> <p>Source of Inspiration: Scaled Agile Framework Glossary</p> |

- New icons for Strategy Definition and Strategy Execution packages
- The shape of the Entity element slightly has been changed to easily differentiate with any other type of element

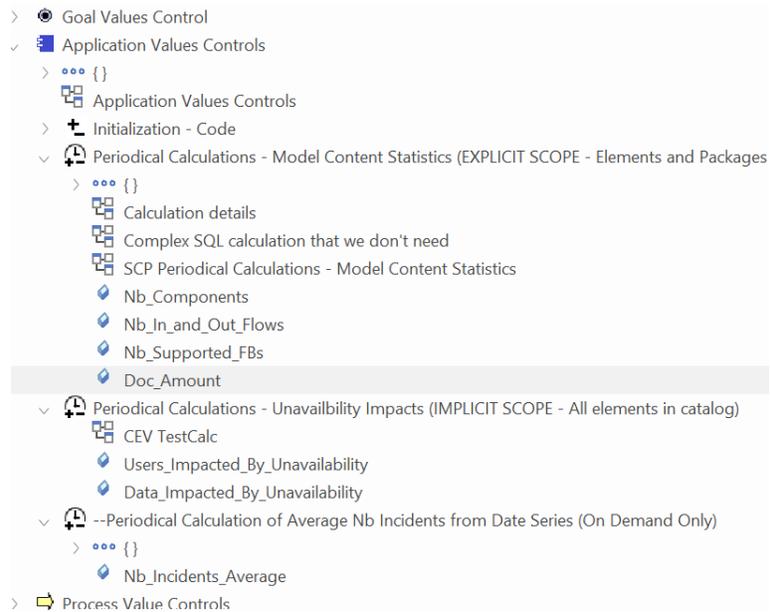


- New "Chart" option for the "Format" property of Representation elements, accompanied by a matching icon to distinguish its shape.



(Cascaded) Calculation

- The sequence of properties calculations always follows the sequence of property calculations defined in the model structure (visible in the "project browser" window). This now also applies to calculated properties defined inside a same "Periodical Value Calculation Rule". This is important for cascaded calculations where one property calculation uses the results of previous property calculations.



- Fixed issue when a property name is a substring of another property name in the same calculation.
- New shorthand operators to calculate minimum and maximum values. See "Mathematical Expressions" in the [calculation documentation](#).

Standard Metamodel

- Added connection
 - Organization - Is part of -> Organization

B0601

Sparx Systems' EA 64 bits and SQLite (QEA/QEAX) database support

- Labnaf software is now provided in 64 bits format.
- Local database files
 - SQLite (QEA) database files replace Access database files since Access is no longer supported by Sparx EA 64 bits.
 - Backup from SQL Server to database file (now QEA file) is no longer limited in size
 - All sample Access database files have been replaced by QEA files

Language

- New or improved **Strategy Definition** viewpoints with [template folders/packages and template diagrams](#) for automatic layout and pre-fill. The templates can be customized. Viewpoints come with sample views.
When a new diagram of any of the following types is created the template automatically applies.

You can copy these templates either from the Labnaf sample repository or from the startup repository into your own repository.

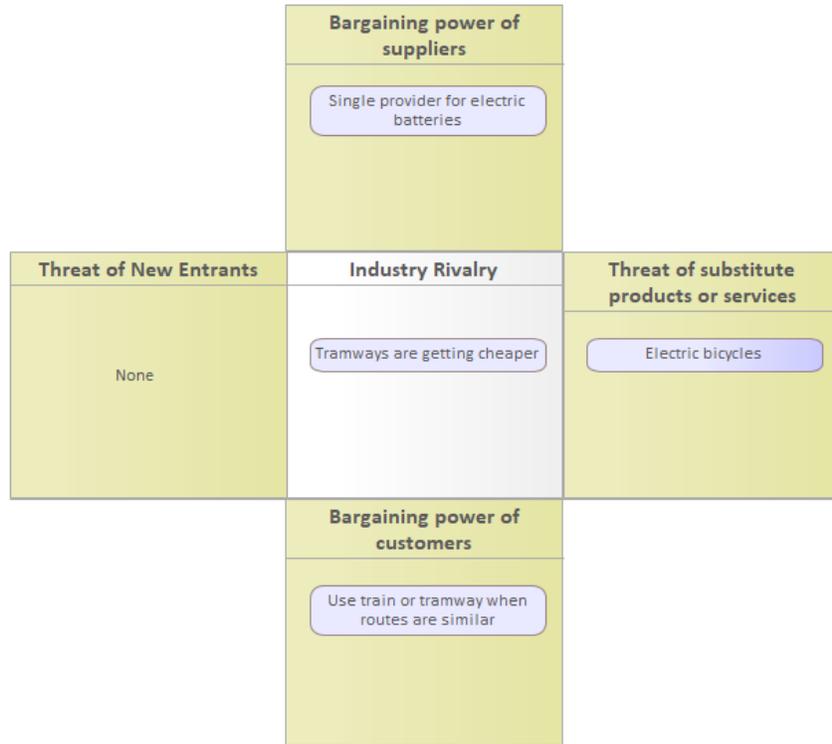
- [Enterprise Ecosystem](#)

Context Analysis & Diagnoses views are populated with [Assessment](#) elements

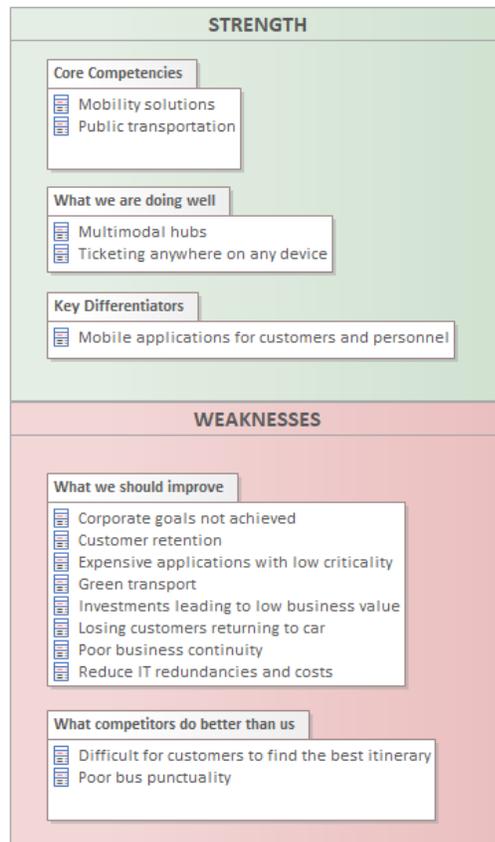
- [Macro-Environment \(PESTEL\)](#)

| Political | Economical | Socio-cultural |
|---|---|---|
| <ul style="list-style-type: none">Mobility strategy and priorities are defined by governmentTowards multi-modal transportTowards multi-modal hubsMigration to electric shuttles | <ul style="list-style-type: none">Decreasing public funding | <ul style="list-style-type: none">Growing number of digital nativesGrowing awareness regarding sustainable mobility |
| Technological | Environmental | Legal |
| <ul style="list-style-type: none">Mobile applications for multi-modal route planning and paymentCheaper and more efficient electric vehiclesTowards self-driving cars and shuttlesDeployment of public infrastructure for optimizing traffic fluidity: smart traffic lights, dynamic road signs, sensors and cameras etc | <ul style="list-style-type: none">Growing air pollution in Syldania | <ul style="list-style-type: none">Employees benefits from car allowance to public transportCash for carWork week regulations in public transportSafety regulations |

- Competitive Environment (Porter's Five Forces)



- Internal Analysis & Diagnoses



Any additional kind of dashboard, chart, matrix, report or free text.

- [SWOT](#)

| | Internal  Details | External  Details |
|---------|--|---|
| Helpful | <p style="text-align: center;">STRENGTH</p> <ul style="list-style-type: none"> Mobility solutions Public transportation Multimodal hubs Ticketing anywhere on any device Mobile applications for customers and personnel | <p style="text-align: center;">OPPORTUNITIES</p> <ul style="list-style-type: none"> Automated Identification & Governance of Assets at Risk Improve punctuality by connecting to new public infrastructure optimizing traffic fluidity Leverage mobile infrastructure to optimize customer loyalty and to attract new customers |
| Harmful | <p style="text-align: center;">WEAKNESSES</p> <ul style="list-style-type: none"> Investments leading to low business value Corporate goals not achieved Poor business continuity Expensive applications with low criticality Poor bus punctuality Losing customers returning to car Difficult for customers to find the best itinerary | <p style="text-align: center;">THREATS</p> <ul style="list-style-type: none"> Unknown assets at risk Undefined usage of sensitive information Decreasing public funding needed to adapt to sustainability and smart mobility |

- Strategic Foundations
 - [Value Proposition Canvas](#)
 - [Business Model Canvas](#) (improved)
- Epic
 - New tagged value
 - Value_Delivery_Score (0-5)
- Process and Value Streams
 - Can use function flows
 - Functional Flows can carry Value elements
- Functional interactions (Enterprise Functions / Business Capabilities interactions)
 - Functional Flows can carry Value elements

Standard Metamodel

- Added connections in metamodel
 - Process (including value stream) – Has information flowing to -> Process
 - Application - Is part of -> Product (for organizations delivering software products)

Import Tabular Report Dialog

- Identifies stereotype from first Excel sheet name (if the file is an Excel file)

Calculations

- New Expected_Unavailability of Processes and Activities
 - = MAX (Expected_Unavailability of all applications supporting all embedded processes and activities including self)

Labnaf AddIn Lite

- Instant Metadata Manager is now enabled

B0507

Connect Elements (no diagram needed)

Using the "Connect Elements" window, you can quickly create connectors without having to create a diagram or a matrix.

Select two elements from the browser window or from diagrams, and the Connect Elements form will then show you which connections are valid according to the active metamodel.

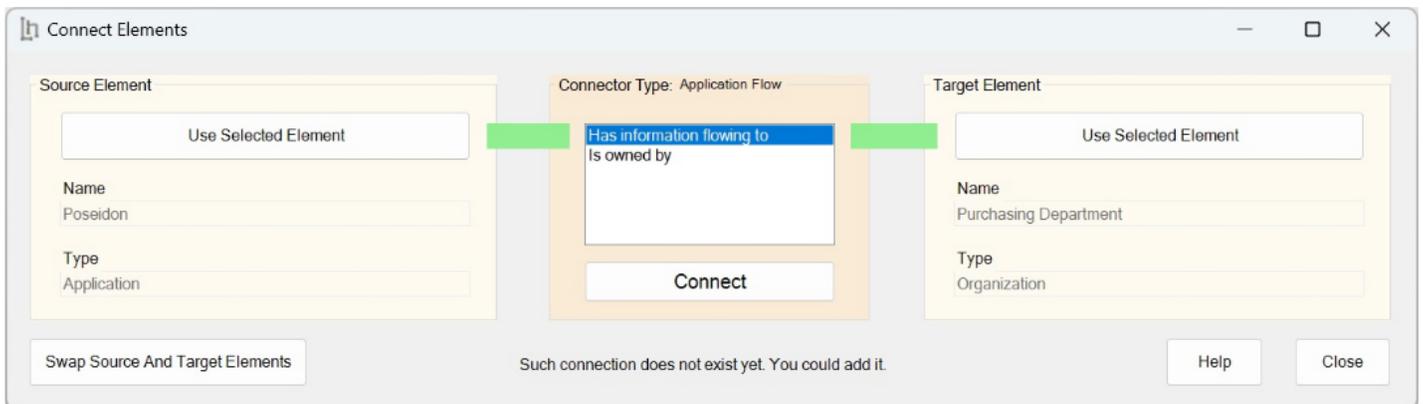
The connection direction is important. So, you can click on the "Swap Source And Target Elements" if necessary.

The window also shows you which connectors, among these valid connectors, already exist between the two elements.

You can add connections that don't exist yet, or you can delete existing connections.

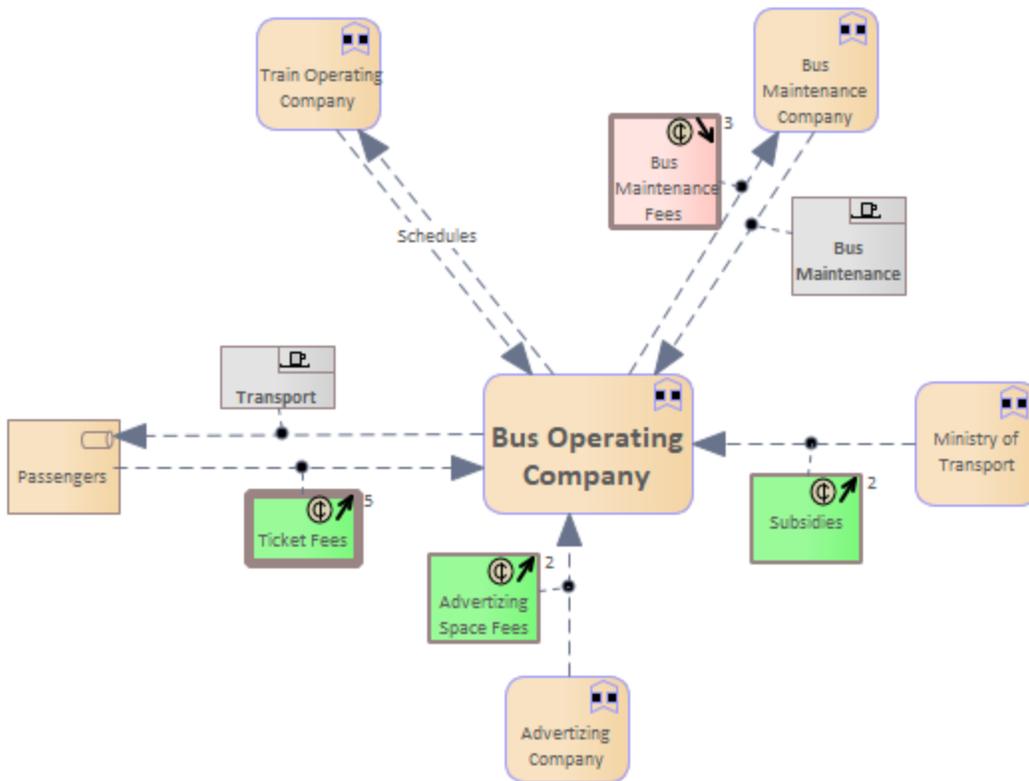
HOW TO ACCESS

1. Right click either on a source element to be (dis)connected or anywhere in a diagram or in the browser window
2. Select **Specialize > Labnaf > Connect Elements**
3. **Select the source and target elements** from the browser window or from diagrams
=> The window shows the list of applicable connector types between these elements.
The list of available connector types is generated by the active metamodel.
4. **Select a connector type or double-click to connect immediately**
Anyone might be sometimes absent-minded. Therefore double-click never deletes existing connections
- 5a. If the connection does not exist, a button labeled '**CONNECT**' will let you create a connection from the source element to the target element.
- 5b. If the connection already exists, a button labeled '**DISCONNECT**' will let you delete the connection from the model.



Language

The **Context Analysis and Diagnosis** viewpoints can now include **enterprise ecosystems**.



Standard Metamodel

- Added connections in metamodel
 - Capability – Aggregation (Is Part Of) -> Product
 - Capability – Impact -> Contract

Canonical Repository Structure

A new version of the canonical repository structure is available [here](#).

Empty Repositories

When creating connectors in an empty repository, a Labnaf configuration package is no longer automatically created.

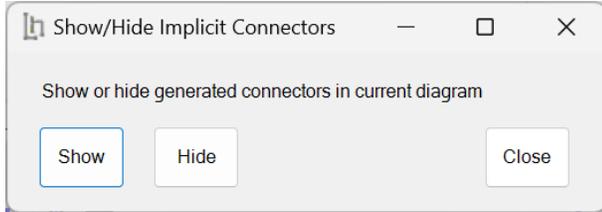
Language Transformer

Updated examples and documentation

Implicit Data Management

Hide/Show Generated Connectors

- Ability to quickly hide or show generated connectors in the currently active diagram.
- You can keep this little window open and open new diagrams to hide or show generated connectors.



Additional Generation of Entities Aggregations (indicating information usage)

- Entities used by roles when roles realize swimlanes that directly or indirectly use these entities
- Entities used by logical nodes when the entities are directly or indirectly accessed by data stores deployed on the logical node via an application deployment set.
- Entities exchanged between application components via the exchanged representations aggregating entities

Tabular Report Generation and Import

- Sparx EA built-in "Priority" property is now supported as well.
- Custom report generation works also for Information Security Requirements (no more message saying worksheet name is too long).

Validation

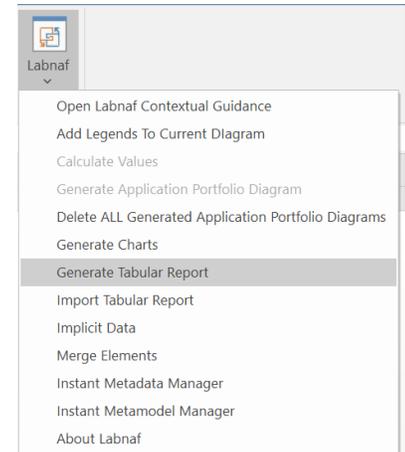
- When a user tries to use a UML connector to connect Labnaf elements (which is not allowed), the error message tells it was a UML connector that was requested.

Miscellaneous

- New Search: Connectors – All Labnaf Connectors

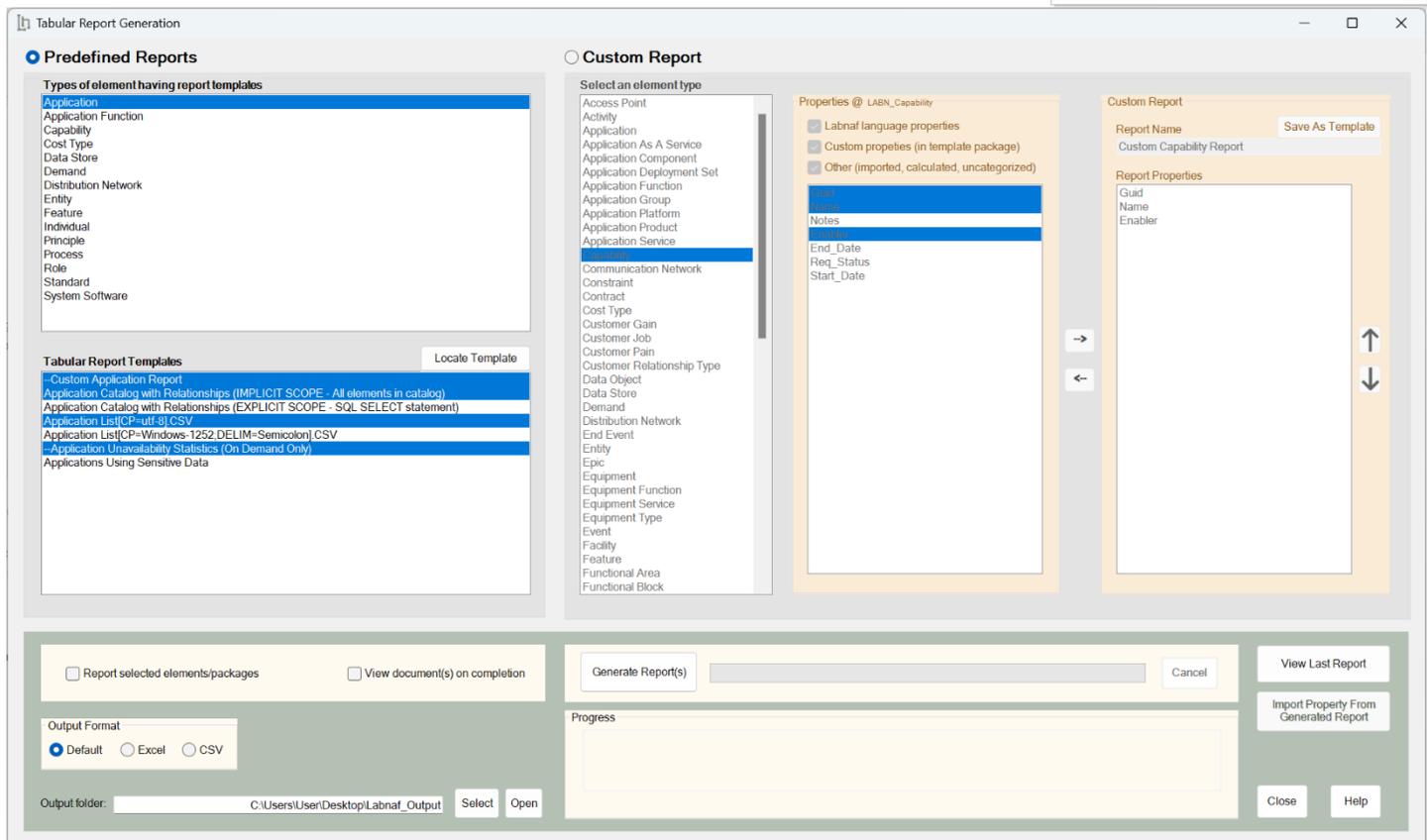
Ribbons

The Labnaf button now appears on the “Start” ribbon (last button on the right).



Tabular Report Generation (Excel & CSV)

New Features and New User Interface



Using Predefined Report Templates

- Select an element type that has defined report templates
- Select tabular report templates
- Select an output folder (typically only once).
- Select an output format (Excel or CSV) i.e. not necessarily using the format defined by the templates.
- Generate reports for a selection of elements (scope) of your choice (the sequence below corresponds to priorities)
 - for elements or packages selected in the project browser,
 - or for elements or packages selected in the active diagram,
 - of for elements and packages present in a “scope” diagram that belongs to each template,
 - or based on a specific SQL select defined in each template,
 - or for the entire catalog (default).
- View the generated reports, or just the last one.

- Open the output folder to see all of generated reports.
- Open the “Log” folder that is located just underneath.
- Locate the selected template in the repository in case you want to make some changes, or if you want to clone and adapt it.

Creating Custom Reports and Generating Templates

- Create a custom report for a specific element type and for selected properties
- Easily reorder properties
- Generate custom report
- Open generated Excel or CSV, edit values, and import the updates values on the fly.
- Save custom report as template
- Locate and edit the template to extend its content, colors or to change the scope of elements to be processed

Quick update of same property for selected elements or packages

Labnaf PowerShell

CmdGenerateTabularReport (using Labnaf PowerShell)

- The “ElementPrototype” parameter can now be a specific element prototype name, an element stereotype or a friendly stereotype name (e.g. 'Application') for which all or specific tabular report templates must be applied.
- More detailed logging info provided.

CmdValidate

- The command now detects and reports duplicate connectors for the connector types that must be unique between same source and target elements.

Logging Files

- All commands yield a specific log file (in addition to standard output).
Each log file name includes a time stamp, the name of the command, and additional context when applicable.
The log directory path is specified by the “OUTPUT_DIR” environment variable. By default, the provided Labnaf PowerShell scripts set the following value to this variable: “...Labnaf_PowerShell\SCHEDULED\Output”.
If the environment variable “OUTPUT_DIR” is not set then log files are created in “...\Desktop\Labnaf_Cmd_Output\Log”.
- New scripts “Cleanup_LogFiles.cmd” and “Schedule_Cleanup_LogFiles.cmd” delete log files older than a certain number of days. By default, the script sets the number of days to 30. Integrated in the global scheduling.

Scope (of processing)

Excel/CSV Report Generation, Calculations, and Chart Generation can apply to

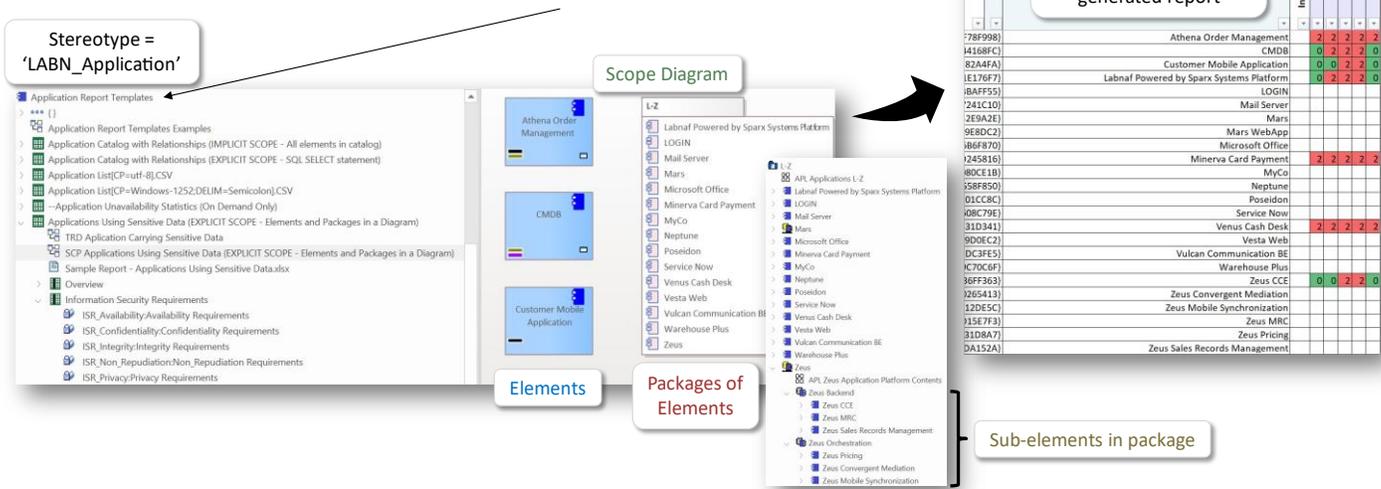
- a specific type of element or a specific report template
- and on
 - Elements or packages selected in the project browser or in the active diagram
 - Elements and packages present in a “**scope**” diagram that belongs to the template
 - Elements selected by a SQL select defined in the template note
 - All elements in the catalog (default)

If multiple scopes have been defined, then the above sequence corresponds to priorities

Populating a scope diagram

Add the elements and packages to be reported to the scope diagram
=> the collected elements are

- Elements and packages of elements (including sub-elements) in the scope diagram
- that have the same stereotype as the parent element prototype



Metamodeling & Model Validation

Preventing duplicate connections in the metamodel

- Adding duplicate metamodel connections is not permitted. This now also applies to the currently disabled metamodel.

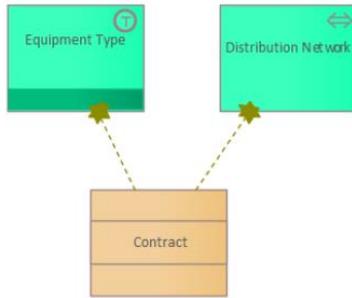
Preventing duplicate connections for specific types of connectors

- The user interface now prevents you from creating **new** duplicate connectors. This is applicable for the configurable list of connectors that must be unique between same source and target elements.
- The CmdValidate command now detects and reports **existing** duplicate connectors for the connector types that must be unique between same source and target elements.

Generating a diagram documenting your customization of the standard metamodel

The generated diagram now includes:

- the custom metamodel connectors (as before),
- the configurable list of connectors that must be unique between two same elements,
- the verbs used in quick linkers.



The present diagram shows the connectors that were added to the standard metamodel along with the customizable lists of unique connector types and quick link verbs.
 These lists are stored in tagged values of the package that contains the metamodel elements and connectors.

LIST OF CONNECTOR TYPES THAT MUST BE UNIQUE BETWEEN TWO SAME ELEMENTS

- Stereotype
- LABN_Access
- LABN_Assignment
- LABN_ContractualBinding
- LABN_Dependency
- LABN_Deployment
- LABN_Evolution
- LABN_Fit
- LABN_Generalization
- LABN_Impact
- LABN_Influence
- LABN_InstantiatedDeployment
- LABN_InstantiatedRealization
- LABN_Node to Network Connection
- LABN_Realization
- LABN_Triggering
- LABN_UsageRestriction

- VERBS USED IN QUICK LINKERS
- Stereotype, VerbTo, VerbFrom
 - LABN_Access, Access, Access
 - LABN_Aggregation, Is part of, Aggregates part
 - LABN_ApplicationFlow, Has information flowing to, Has information flowing from
 - LABN_Assignment, Is assigned to, Has assignment from
 - LABN_Association, Is associated with, Is associated with
 - LABN_ComponentFlow, Has information flowing to, Has information flowing from
 - LABN_Composition, Is a constituent of, Is composed of
 - LABN_ContractualBinding, Is bound to contract, Has contractual organization party
 - LABN_Dependency, Depends on, Needed by
 - LABN_Deployment, Is deployed on logical node, Serves
 - LABN_Evolution, Evolves as, Evolves from
 - LABN_Fit, Fits, Fits
 - LABN_Flow, Has resources flowing to, Has resources flowing from
 - LABN_FlowAllowedByFirewall, Allows flow, Is allowed by firewall
 - LABN_FunctionalFlow, Has entities flowing to, Has entities flowing from
 - LABN_Generalization, Specialises, Generalizes
 - LABN_Impact, Impacts, Is impacted by
 - LABN_Influence, Influences, Is influenced by
 - LABN_InstantiatedAggregation, Is part of node instance, Aggregates
 - LABN_InstantiatedDeployment, Is deployed on node instance, Serves
 - LABN_InstantiatedRealization, Realizes logical node, Is realized by logical node
 - LABN_NetworkInterlink, Has link with, Has link with
 - LABN_Node to Network Connection, Is connected to, Has connection from
 - LABN_Ownership, Is owned by, Owns
 - LABN_Path, Has path to, Has path from
 - LABN_PhysicalFlow, Has material or information flowing to, Has material or information flowing from
 - LABN_Realization, Realizes, Is realized by
 - LABN_Triggering, Triggers, Is triggered by
 - LABN_UsageRestriction, Restricts usage of, Has usage restricted by

Updated Sparx EA User's Permissions Enabling Labnaf Features

| Labnaf Feature | Required by Sparx EA | Required by Labnaf |
|--|------------------------------|------------------------------------|
| Calculate Values | 10 Update Elements | 6 Manage Replicas |
| Generate Diagram and Chart | 10 Update Elements | 6 Manage Replicas |
| | 4 Update Diagrams | |
| | 25 Manage Diagrams | |
| | | |
| Create Diagram Based On Template Package | 10 Update Elements | |
| | 4 Update Diagrams | |
| | 25 Manage Diagrams | |
| Generate Tabular Report | | 23 Generate Documents |
| | | |
| Import Tabular Report | 10 Update Elements | 12 Import XMI |
| | 19 Configure Stereotypes | |
| | | |
| Generate Implicit Data | 10 Update Elements | 6 Manage Replicas |
| | | |
| Merge Elements | 10 Update Elements | 6 Manage Replicas |
| | 4 Update Diagrams | |
| | 25 Manage Diagrams | |
| | | |
| Instant Metadata Manager | 10 Update Elements | 19 Configure Stereotypes |
| | | 38 Configure Project Prerequisites |
| | | |
| Instant Metamodel Manager | 10 Update Elements | 19 Configure Stereotypes |
| | 4 Update Diagrams | 38 Configure Project Prerequisites |
| | 25 Manage Diagrams | |
| | 33 Baselines - Manage | |
| | 34 Baselines - Restore model | |

Miscellaneous

- Work around for Sparx EA bug when adding elements using the "Add Element" from a package

Labnaf provides two options for language customization.

[Related Labnaf Guidance entry](#)

OPTION1 (new): Use the **Instant Metamodel Manager** and **Instant Metadata Manager** provided by the Labnaf AddIn. They are directly part of the modeling environment.

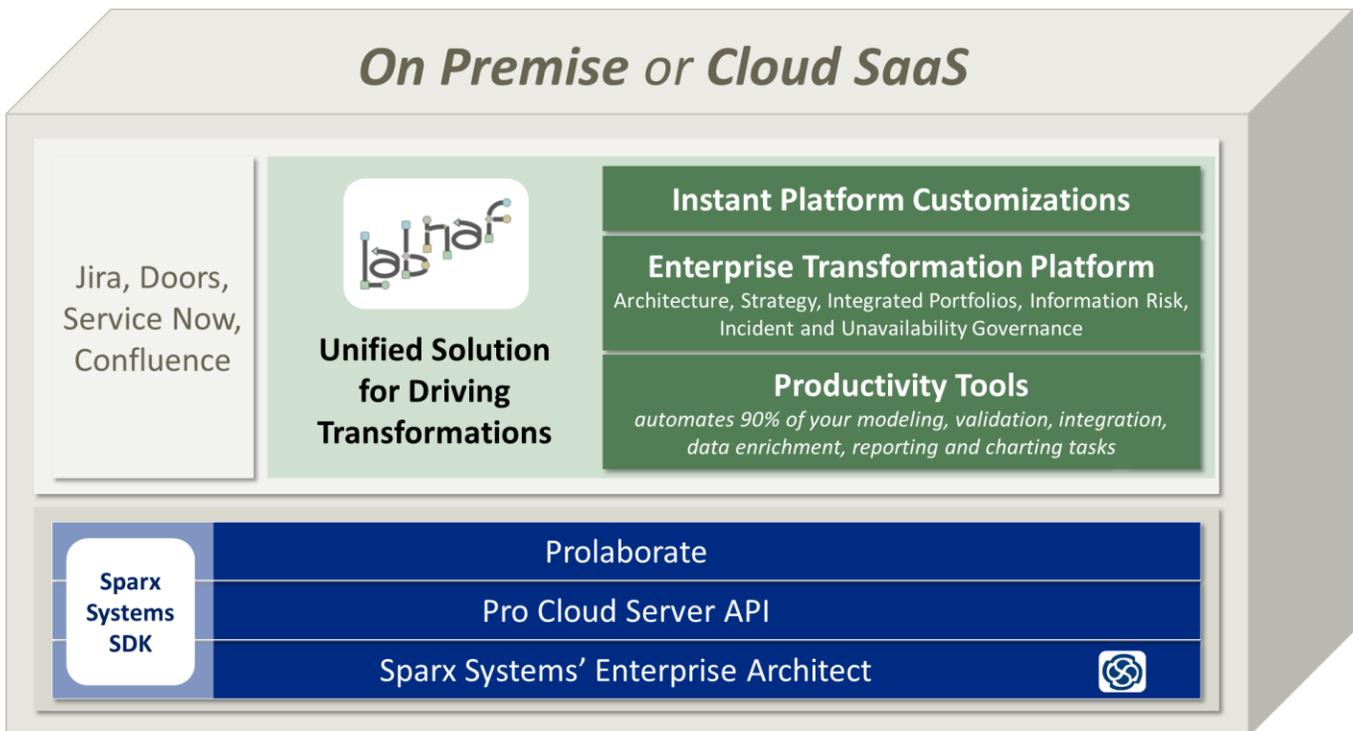
Using these tools, you can decide to

- use the standard built-in metamodel, customize it, or create your own user-defined metamodel from scratch,
- visualize, create, rename, delete and synchronize element properties,
- automatically merge new versions of Labnaf with your own metamodel and metadata (properties) customizations.

Because the Instant Metamodel Manager and Instant Metadata Manager are available directly in the modeling environment, you don't need any extra tool.

The modeling configuration (MDG) is dynamically updated for you behind the scene.

So, this the perfect option for modeling in an **Enterprise Architect SaaS environment (cloud)**.



OPTION 2: Customize the Labnaf MDG and generate the connection constraints using the Customization Workbench

This is for advanced customizations.

Instant Metamodel Manager

[Related Labnaf Guidance entry](#)

Labnaf comes with a standard metamodel that spans the entire process of driving transformations.

There is also a user-defined metamodel that you can populate and activate.

You can easily switch between standard metamodel, customized metamodel, and user-defined metamodel in two clicks.

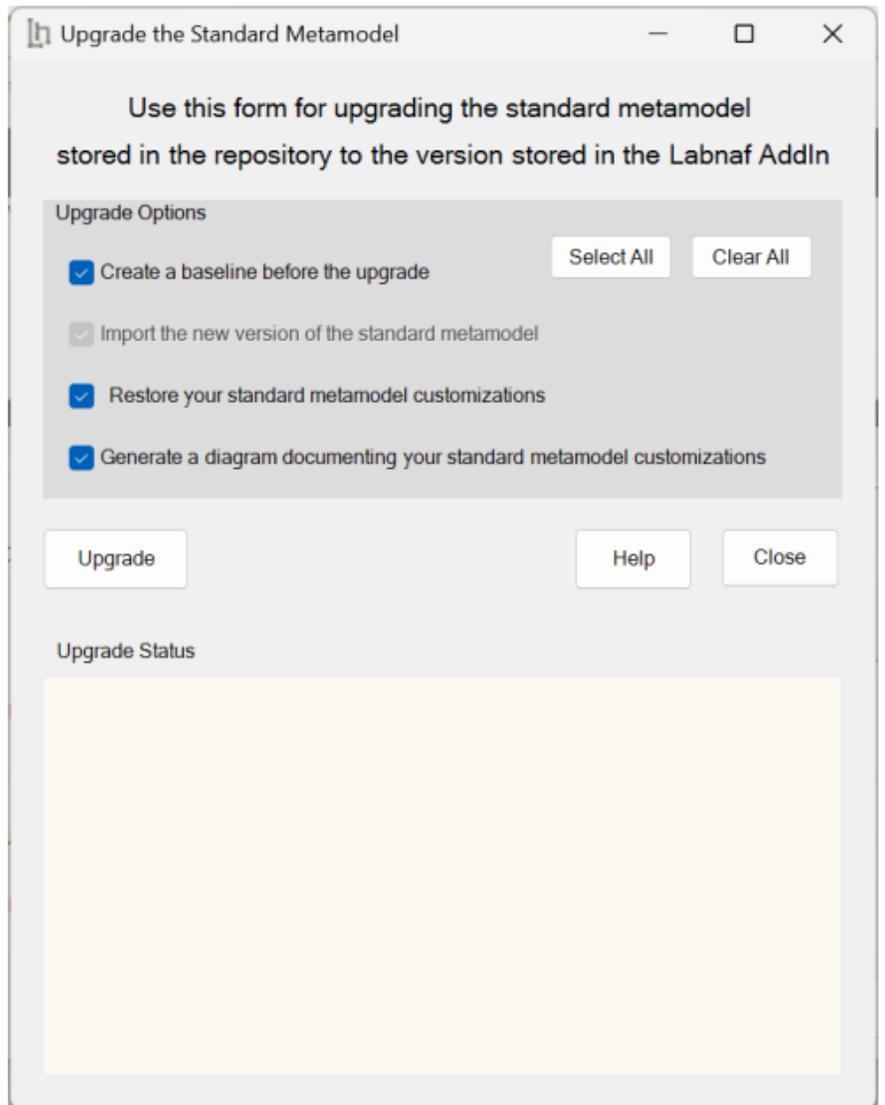
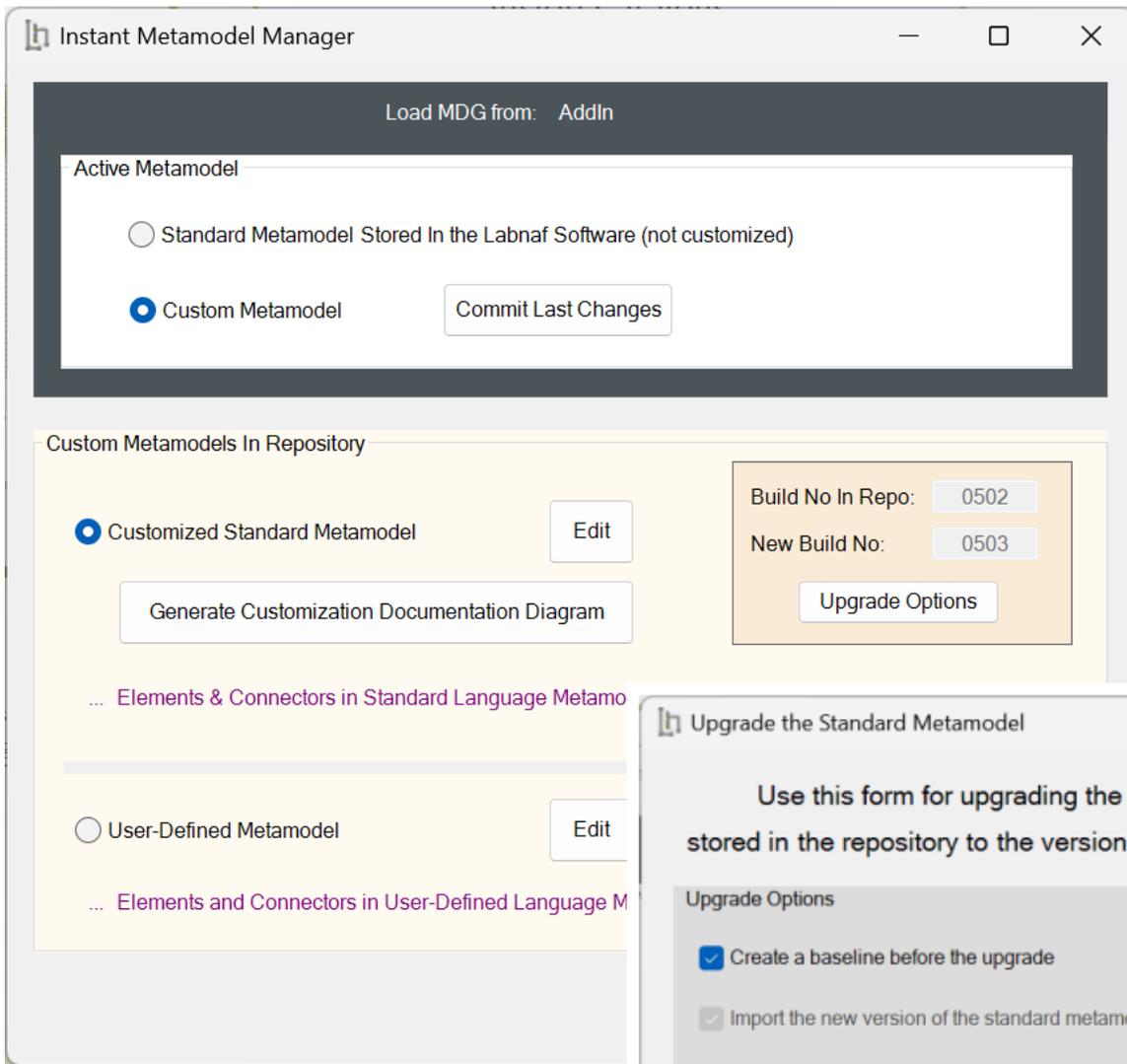
The configurable metamodels are expressed in the end user Labnaf language itself. So, the human-readable metamodel specification and the documentation are one and the same thing.

The metamodels can now be instantly and dynamically updated and upgraded, on premise or on the cloud, using the Instant Metamodel Manager.

Using the Instant Metamodel Manager, you can...

- Select and customize your preferred metamodel (standard, customized standard or user-defined),
- Upgrade the standard metamodel to a new version of Labnaf while keeping your own metamodel customizations,
- Restart an interrupted upgrade,
- Create baselines before the upgrades,
- Generate documentation diagrams on your customization of the standard metamodel on demand and automatically after upgrade,
- Create, upgrade and automatically repair existing metamodel structures (even empty ones) in any repository.
- Manage metamodels using the end user modeling language itself (no metamodeling language needed) and without the need of any MDG models and tools.
- Manage metamodels in an EA SaaS Cloud environment.

| Labnaf Customization Tools => | Labnaf AddIn: Instant Metamodel Manager | Labnaf AddIn: Instant Metadata Manager | Customization Workbench |
|--|--|---|----------------------------|
| Categories of Customization | | | |
| Metamodel | <i>Customizable metamodel expressed using the Labnaf Language and stored in the production database.</i> | | |
| Select active metamodel (standard, customized standard, or user-defined metamodel) | Y | | |
| Add/Delete connectors to a metamodel | Y | | |
| Upgrade standard metamodel to a new version of Labnaf, while keeping your own customizations | Y | | |
| Generate documentation/diagram about your metamodel customization | Y | | |
| | | | |
| Element Properties | | | |
| Create custom property types | | Y | Y |
| Upgrade properties to a new version of Labnaf and keep/restore your own customizations | | Y | |
| Rename/Delete property types | | Y | Y |
| Synchronize property sets in existing elements | | All props | MDG props |
| Add custom properties to the Tags Tab | | Y | N |
| Add Labnaf Properties to the Element Tab | | N | MDG-defined |
| Create named property groups | | N | MDG-defined |
| | | | |
| Connector Properties | | | |
| Create custom connector properties | | N | MDG-defined |
| | | | |
| Elements and Connectors | | | |
| Add/Delete element and connector types | | N | MDG-defined |
| | | | |
| Toolboxes and Diagram Types/Viewpoints | | | |
| Add/Update/Delete Toolboxes | | N | MDG-defined |
| Add/Update/Delete Diagram Types/Viewpoints | | N | MDG-defined |
| Change Shapes | | N | MDG-defined |
| Change Icons | | N | MDG-defined |



Labnaf PowerShell

New command: “**SetProperty**” that selects any elements or packages and set some property value.

Language

- Added stereotypes for existing and new metamodel packages.

Standard Metamodel

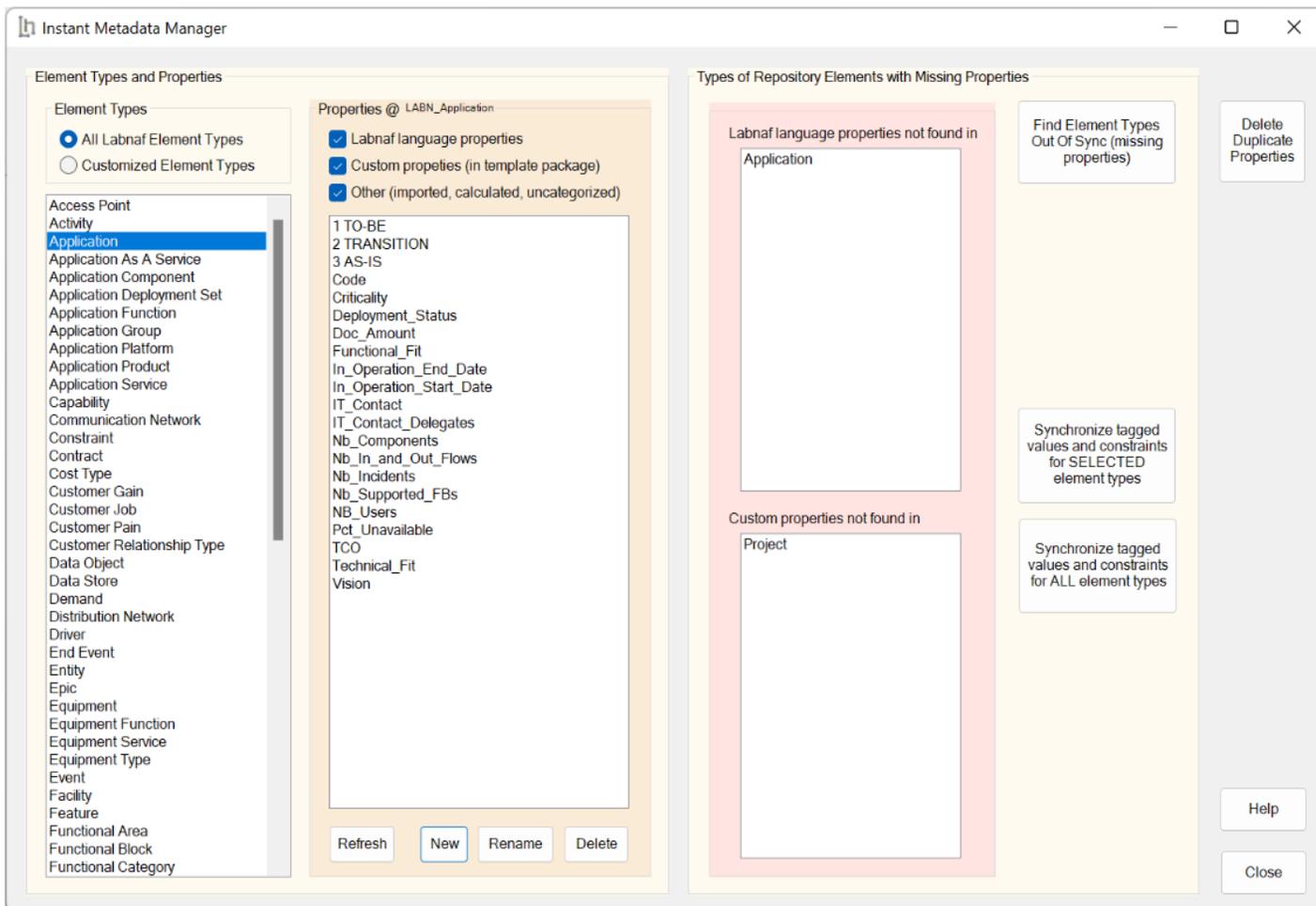
- Added connections in metamodel
 - Epic – Realization -> Strategic Theme, Strategic Objective
 - Project – Realization -> Demand

Miscellaneous

- Application Portfolio Diagram Generation does no longer show error when there is no application in the catalog.

Use the Instant Metadata Manager to instantly manage custom properties that live independently of any future version of the Labnaf language.

So there is no upgrade issues.



Using the Instant Metadata Manager, you can...

View the list of existing element types.

View a filtered list of element properties for the selected element type. Using the filters you can include or exclude

- The element properties defined in the Labnaf language installed on your computer. The Labnaf language evolves over time and you might also have branched your own version of the Labnaf language using the Customization Workbench.
- The custom properties that you have added using the Instant Metadata Manager.
- Other properties that were added to elements as a result of data import and value calculations. It could be also properties that you added to specific elements in the user interface or by using your own custom code/scripts.

Add, rename and delete custom element properties instantly, on the spot (no need to learn and to deal with MDG customization lifecycle).

Add custom properties that reuse your preferred property/tagged value types.

Shows **which types of elements** stored in the database **are out of sync with the metadata definitions.**

Identify **which source of meta data definition the elements in the database are out of sync with.** This sources of metadata can be either

- the custom property types that you defined and that are stored in a "Template Package", or
- the Labnaf language (that you might have customized using the Customization Workbench).

Resynchronize elements stored in the database with their metadata definition. This adds all the missing properties in the elements stored in the database. .

Delete duplicate properties. Indeed, the Labnaf user interface and its API enable users to create multiple "tagged values" (custom properties) with the same name to the same element. As far as Labnaf architecture is concerned, duplicate properties are not useful, and they can lead to weird reporting results.

Inbound/Outbound Content Formatting

Labnaf automatically adapts the format of inbound/outbound CSV file content following your systems integration requirements and constraints.

The CSV file format that you can adapt includes any combination of the following items:

- **Character encoding**
- **CSV column delimiters**

Inbound/outbound content formatting applies to

- **Imported CSV files**
- **Exported/Reported CSV files**

Exported/Reported CSV files can be created

- Using **Tabular Report Templates** (On demand or Labnaf PowerShell)
- Using the Labnaf PowerShell **SqlExportToCSV** command

These features add the element **property name mapping** available with the **Import** feature.

[Import feature description.](#)

How To Use

The **character encoding** and the **CSV column delimiter** is specified in the **CSV file name** or in the **Tabular Report name**.

The syntax is

...[CP={code page number or name};DELIM={COMMA|SEMICOLON|TAB|SPACE|a character code number}]...CSV

Examples:

- Applications [CP=Windows-1252;DELIM=Comma].csv
- Applications [CP=1252;DELIM=59].CSV (where 59 is the character code for semicolon i.e. ";")
- Applications [CP=utf-8;DELIM=Semicolon].csv

Supported Character Encoding Schemes

[What is "Character Encoding"?](#)

To specify the character encoding you can use either the **code page number** or the **code page name**. Labnaf will accept values present in the following table:

| Code Page | Name | Description |
|-----------|--------------|-----------------------------|
| 1200 | utf-16 | Unicode |
| 1201 | unicodeFFFE | Unicode (Big endian) |
| 1252 | Windows-1252 | Western European (Windows) |
| 12000 | utf-32 | Unicode (UTF-32) |
| 12001 | utf-32BE | Unicode (UTF-32 Big endian) |
| 20127 | us-ascii | US-ASCII |
| 65000 | utf-7 | Unicode (UTF-7) |
| 65001 | utf-8 | Unicode (UTF-8) |

The default code page is "utf-8".

CSV Column Delimiter

To specify the CSV Column delimiter use one of the following values:

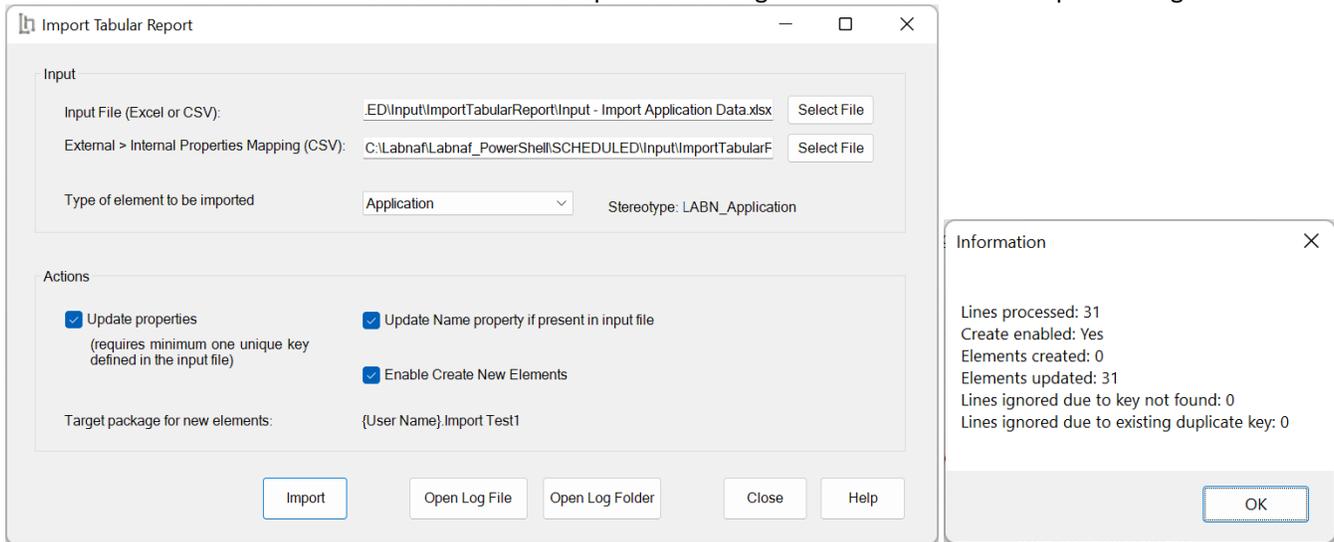
- COMMA (default)
- SEMICOLON
- TAB
- SPACE
- A number that corresponds to the character code.

Labnaf PowerShell

- More detailed processing information provided for import of elements, import of connectors, and document generation commands.

Import elements From Excel or CSV (Labnaf PowerShell or AddIn)

- A textual log file is created in a "Log" folder that is created in the same folder that the one where the imported file is located. The log file name is derived from the imported file name: `..\Log\{Import file name}_Log.txt`
- The AddIn user interface now includes a button to open the last log file and another one to open the log folder.



- Custom properties are created on import (even if the property is not imported)
- Support for multiple character sets and CSV delimiters

Standard Metamodel

- Added connections in metamodel
 - Epic – Evolution -> Project and Program (an approved architecture epic can evolve into one or several implementation projects)
 - Constraint – Influence -> Project and Program

B0501

Labnaf AddIn & PowerShell

- Import tabular report (Excel or CSV) provides a detailed report on achieved operations and supports special encodings.

B0500

90% of architecture and security modeling efforts can be automated. **These are avoidable delays, costs, and human errors.**

Implicit data enable architects to only model what is semantically significant. Labnaf generates the missing content that you need.

Based on minimum modeling efforts, implicit data generation tells you **which individuals, organizations, and roles access sensitive data using what process, detailed activity, function, information flow, application, component, data store, equipment, server and network.**

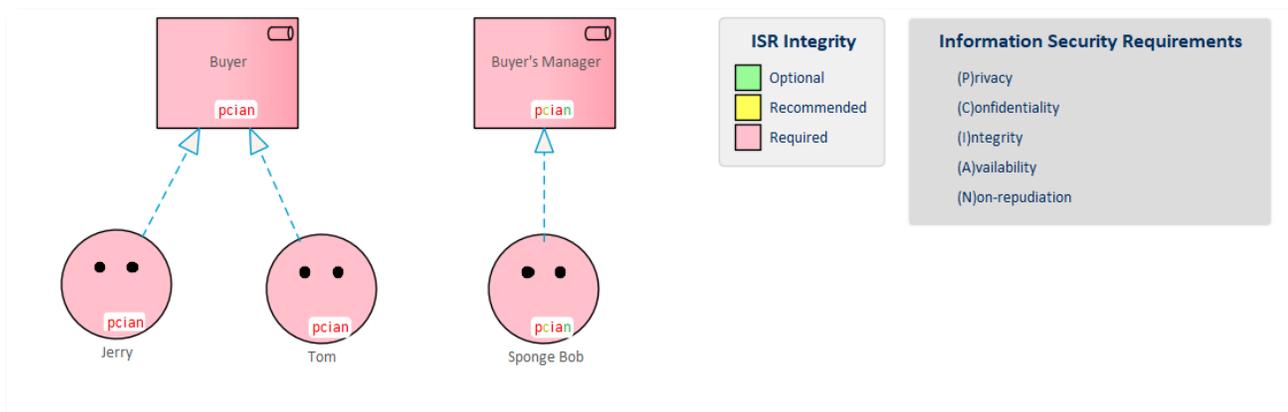
Cascaded calculations (including mathematical expressions, SQL, and time series) further enrich and consolidate data.

From this generated information, **auto-generated and auto-colored diagrams, charts and reports** tell you what levels of **privacy, confidentiality, integrity, availability and non-repudiation requirements** are at stake.

If you also import data about systems unavailability and incidents, this information is further compared and consolidated to provide a **360° view on the impacts of unavailability on users and data access.**

Sensible Data Usage Identification and Consolidation

Automatic detection of who and what has access to sensitive information, and for which specific security requirements. Based on cascaded calculations, implicit data generation, libraries of legends and rich report templates

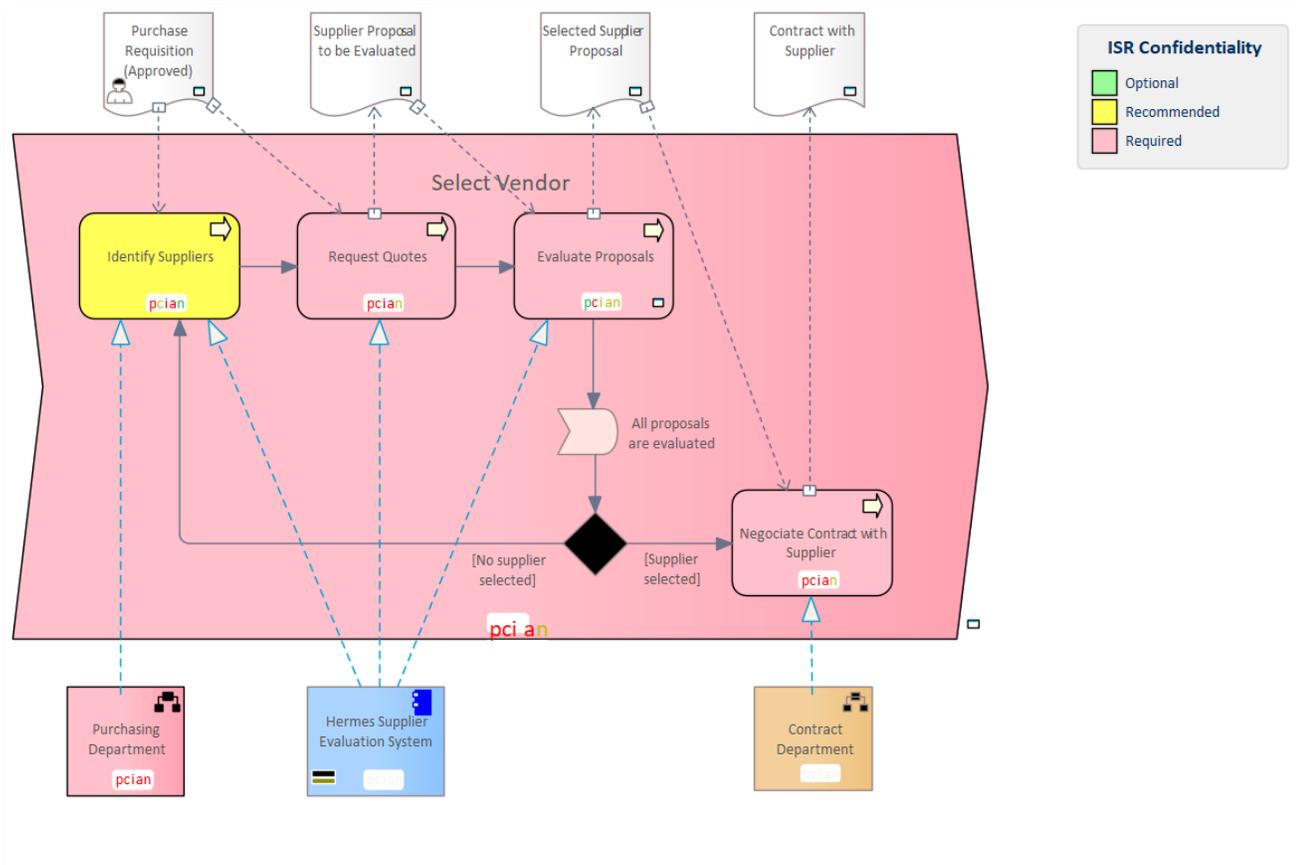


Select your auto coloring legends to focus on specific aspects...

Add Legends To Current Diagram

| Simple Legends | Auto Legends |
|--|---|
| Architecture Perspectives | 1 TO-BE |
| Criticality | 1 TO-BE_ |
| Elements Impacted | 2 TRANSITION |
| Elements New, Changed | 2 TRANSITION_ |
| Elements Under Construction | 3 AS-IS |
| Global Plateaus | 3 AS-IS_ |
| Information Security Requirements | Application Lifecycle / Vision |
| Nb of Applications / Enterprise Function | Average Number of Supporting Applications |
| Roles | Color_Gradation_0_5_Dark |
| TCO | Color_Gradation_0_5_Light |
| | Color_Gradation_0_5_Medium |
| | Consolidated TCO of Supporting Apps |
| | Data Impacted By Unavailability |
| | Differentiators |
| | Externalization |
| | ISR Availability |
| | ISR Confidentiality |
| | ISR Integrity |
| | ISR Non Repudiation |
| | ISR Privacy |
| | KPI Index - Achievement Level |
| | Users Impacted By Unavailability |

Buttons: Add Selected Legends, Clear Selection

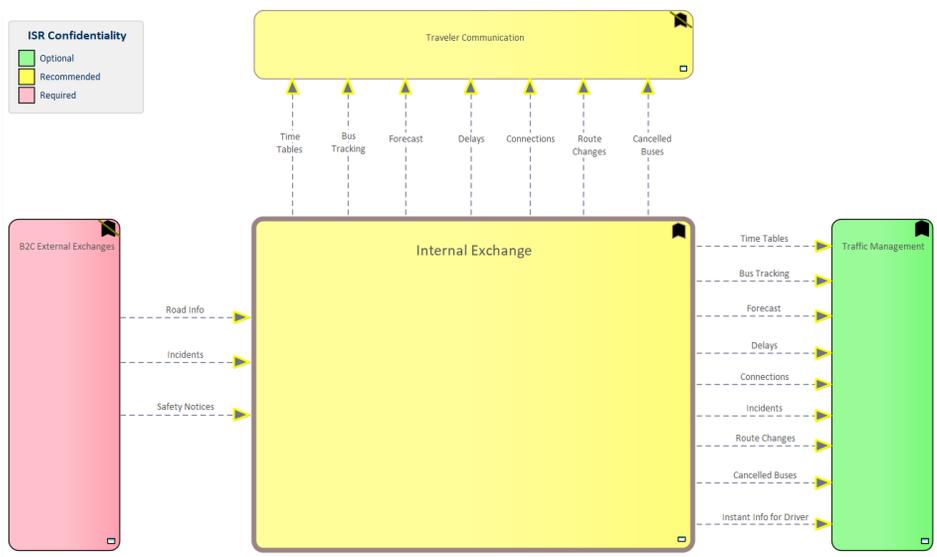


Generate sensitive data usage reports based on template

Zooming in:

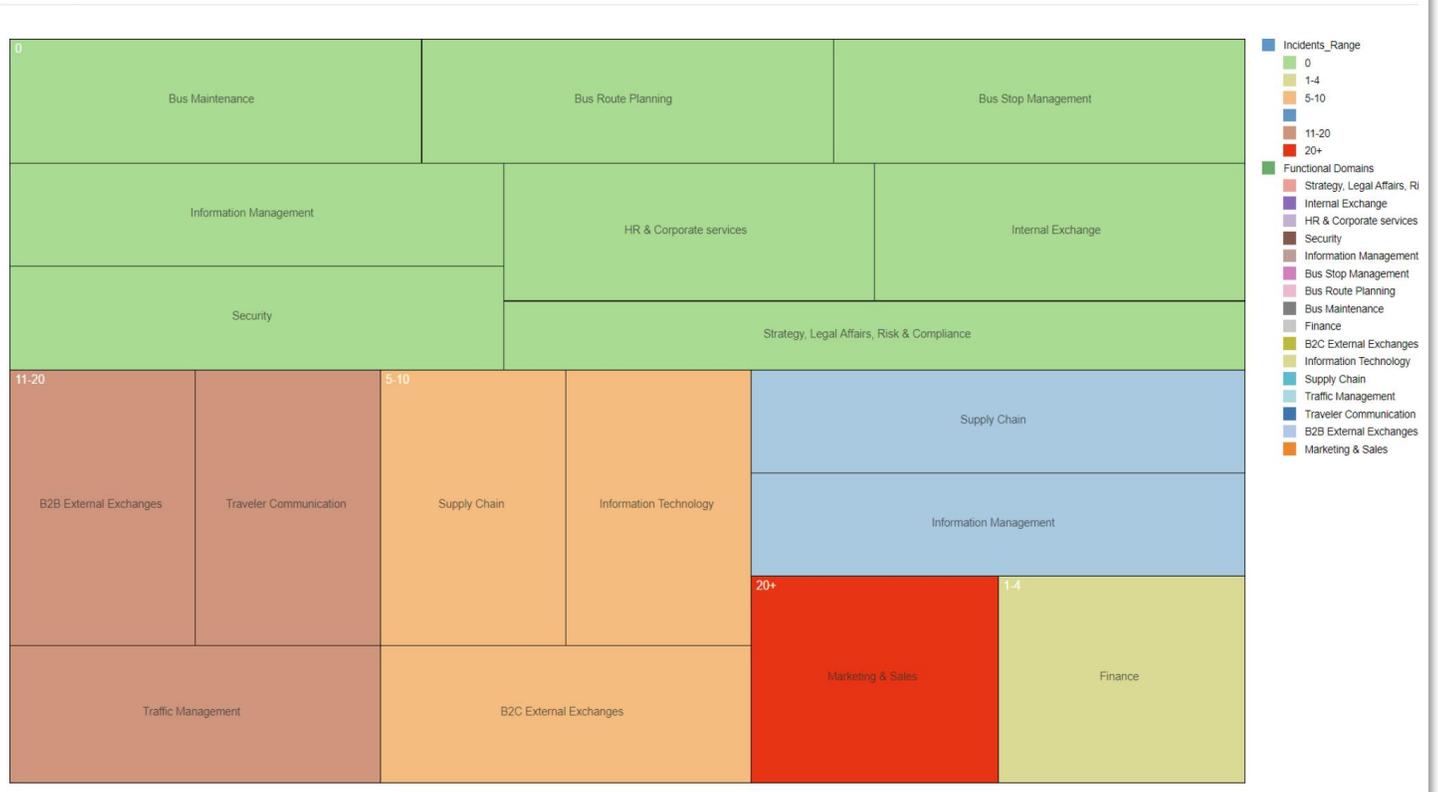
| Name | Security Requirements: | | | | |
|--------------------------------|------------------------|-----------------|-----------|--------------|---|
| | Privacy | Confidentiality | Integrity | Availability | |
| | Non_Repudiation | | | | |
| Account | 2 | 2 | 2 | 2 | 2 |
| Application Architecture Data | 0 | 2 | 2 | 2 | 0 |
| Application Configuration Data | 0 | 2 | 2 | 2 | 0 |
| Applications | 0 | 2 | 2 | 2 | 0 |
| BOPCO Identity | | | | | |
| Bus Planning at D-1 | 0 | 0 | 2 | 2 | 0 |
| Consumer | | | | | |
| Customer | 2 | 1 | 2 | 2 | 0 |
| Customer Order | 2 | 1 | 2 | 2 | 2 |
| Customer Product | 0 | 0 | 2 | 2 | 0 |
| Geography | | | | | |
| Party | | | | | |
| Payment | 2 | 2 | 2 | 2 | 2 |
| Product Price | 0 | 0 | 2 | 2 | 0 |
| Prospect | 2 | 1 | 2 | 2 | 0 |
| Purchase Requisition | 0 | 1 | 2 | 2 | 2 |
| Route | 0 | 0 | 2 | 2 | 0 |
| Subscription Record | 2 | 1 | 2 | 2 | 0 |
| Supplier | 2 | 1 | 2 | 2 | 0 |
| Supplier Product | | | | | |
| Supplier Proposal | 0 | 2 | 1 | 1 | 1 |

Sensitive data usage is automatically consolidated in all directions up to the top-level functional flows and down to the servers and networks.



Incident and Availability Management

Number of Application Incidents in Functional Domains



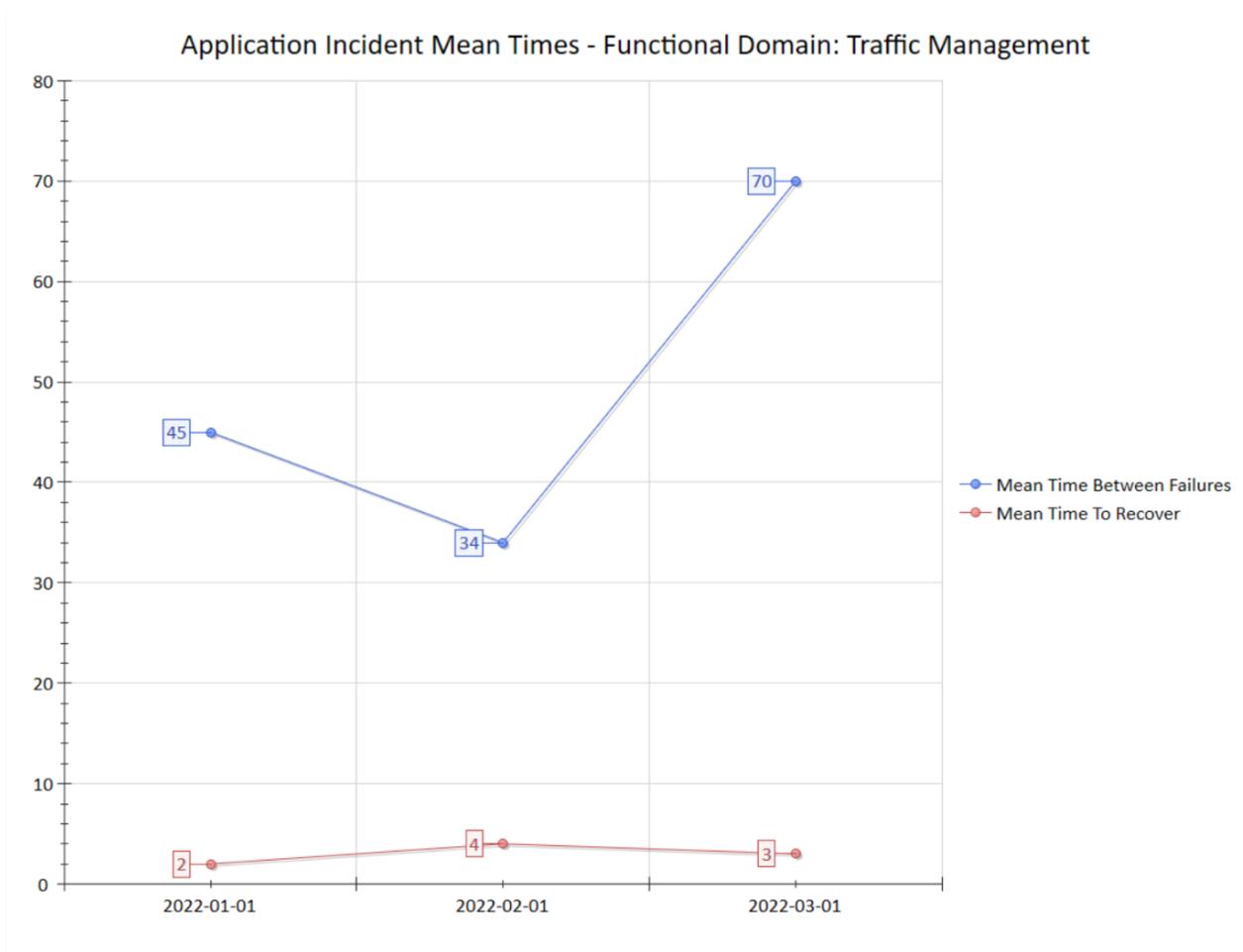
Highest number of application Incidents

Table View

| Functional_Domain | Functional_Area | Functional_Block | Nb_Incidents | Subreport |
|------------------------|----------------------------|---------------------------|--------------|----------------------|
| Marketing & Sales | B2B Sales | B2B Indirect Sales | 19 | View |
| Marketing & Sales | B2C Sales | B2C Order Management | 19 | View |
| B2B External Exchanges | Financial Exchanges | Payment Exchanges | 14 | View |
| Marketing & Sales | B2I Sales | Face-to-Face Distribution | 14 | View |
| Marketing & Sales | B2C Sales | B2C Self-Service Channel | 11 | View |
| Traveler Communication | Pre-Booking Communication | Journey Planner | 10 | View |
| Traveler Communication | Pre-Booking Communication | Journey Tips | 10 | View |
| Marketing & Sales | Multi-channel Sales | Sales Record Management | 9 | View |
| Traveler Communication | Post-Booking Communication | Reminders & Warnings | 9 | View |
| Traffic Management | Customer Services | Disability Assistance | 7 | View |

| Functional_domain | Functional_area | Functional_block | Application | Nb_incidents |
|-------------------|-----------------|--------------------|-------------------------|--------------|
| Marketing & Sales | B2B Sales | B2B Indirect Sales | Athena Order Management | 7 |
| Marketing & Sales | B2B Sales | B2B Indirect Sales | Janus | 6 |
| Marketing & Sales | B2B Sales | B2B Indirect Sales | Demeter | 4 |
| Marketing & Sales | B2B Sales | B2B Indirect Sales | Neptune | 1 |
| Marketing & Sales | B2B Sales | B2B Indirect Sales | Hera | 1 |

Generated Time Series Chart (scheduled or on demand):



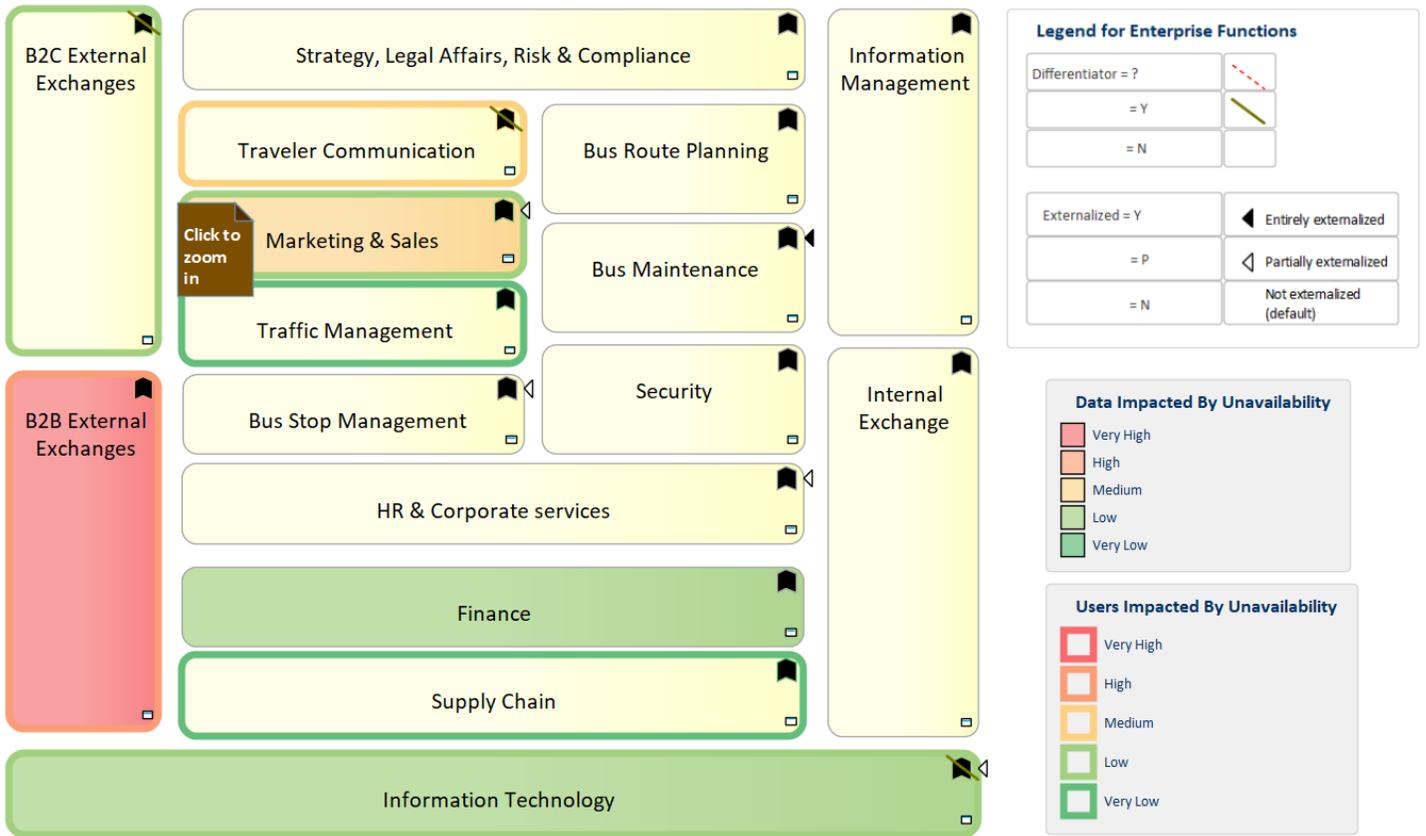
Security and risk management experts have methods and tools (most often disconnected from architecture) to manage risks. It is hard to guess, discuss, or manually evaluate where the risks might be.

Hopefully, Labnaf automatically discovers, consolidates, charts and reports where the security requirements are throughout the architecture.

It also calculates, consolidates, charts and reports the history of incidents.

And finally, it cross-analyses consolidated incident data and availability requirements to produce consolidated heat maps and reports on the impact of unavailability on data access and on users at several levels of detail.

It generates summary data and views, including the evolution over time.



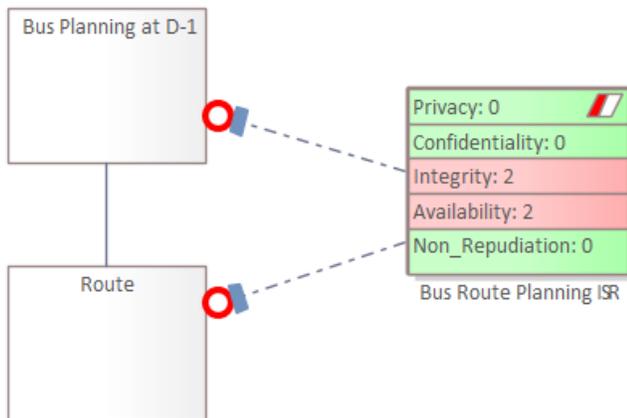
Dashboards

- New dashboards including
 - Application Incident Management
 - Information & Security Requirements
- [New organization of dashboards illustrated in guidance](#) including screenshots from Prolaborate dashboards

Modeling Language Additions

For Information Security Requirements

- **New element stereotypes:** LABN_InformationSecurityRequirement
- **New viewpoint/diagram type and toolbox:** Information Security Requirements
- **Flows of viewpoints:** Added Information Security Requirements viewpoint

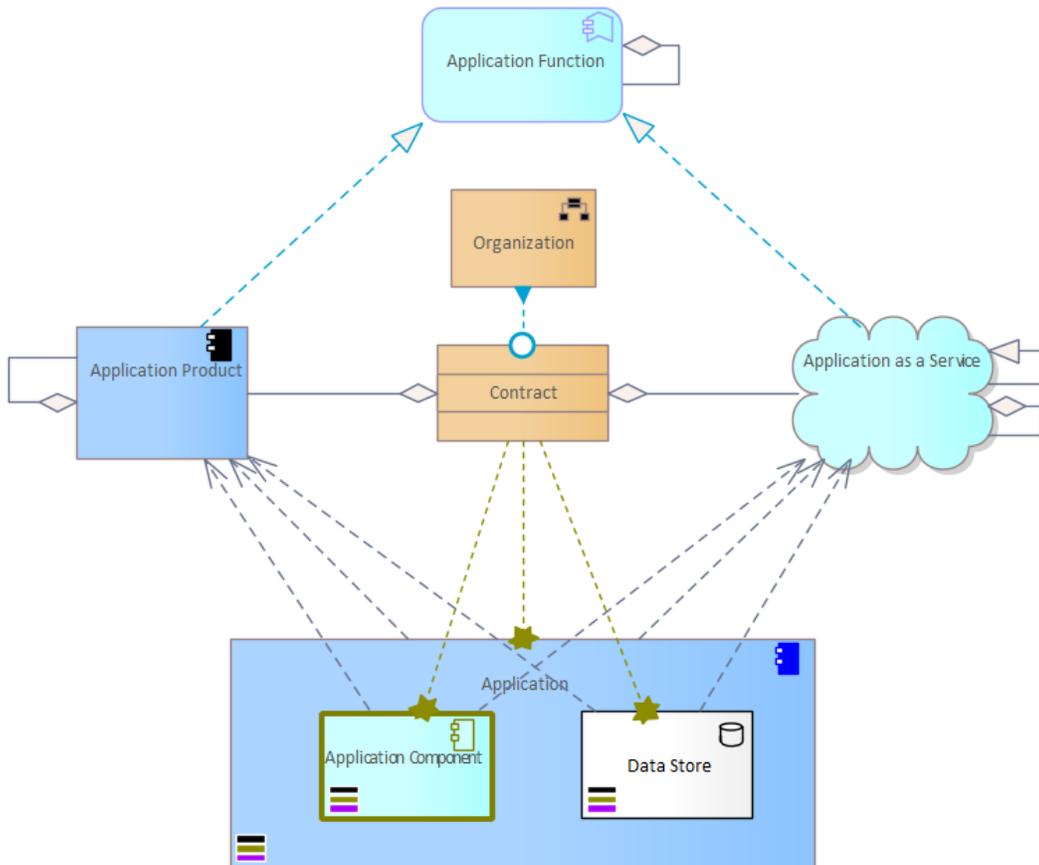


For Incident and Availability Management

- **Application element**
 - New tagged values
 - Nb Incident History, MTBF and MTTR (Time Series)
 - Expected Unavailability
 - User Community Size
 - User impacted by Unavailability: Calculated based on user community size and expected unavailability
 - Data Impacted by Unavailability: Calculated based on calculated sensitive data usage and expected unavailability
 - Sample Excel file for import of above properties
- **Enterprise Function elements (Functional Block, Functional Area and Functional Domain)**
 - New tagged values based on cascaded calculations from Application => Functional Block => Functional Area => Functional Domain
 - Nb Incident History, MTBF and MTTR (Time Series)
 - User impacted by unavailability
 - Data Impacted by Unavailability

For Application Product Management

- New element stereotypes: LABN_ApplicationProduct
- Affected viewpoints and toolboxes: Entities, Contracts and Implications



Library of Legends

- **New Auto coloring legends**
 - Privacy, Confidentiality, Integrity, Availability, and Non-repudiation, Users impacted by unavailability, Data impacted by unavailability
- **New Simple legends**
 - Information Security Requirements
- **Simple Legend Changed**
 - Added CISO to Roles legend

Implicit Data Generation (Scheduled or On Demand)

Used in combination with automatic imports, cascaded calculations, and cascaded chart and diagram generation.

[Details on the Labnaf Guidance web site](#)

Implicit Data

Select All Clear All Save Configuration Update Implicit Data Delete All Implicit Connectors

Implicit Connector Configuration

- Generate aggregations of children by parents

Generate aggregations of Entities by

- Entities Exchangers (Processes, Roles, Applications...)
- Individuals
- Logical Nodes
- Communication Networks
- Parents of all above entities aggregators

Implicit Risk Information Configuration

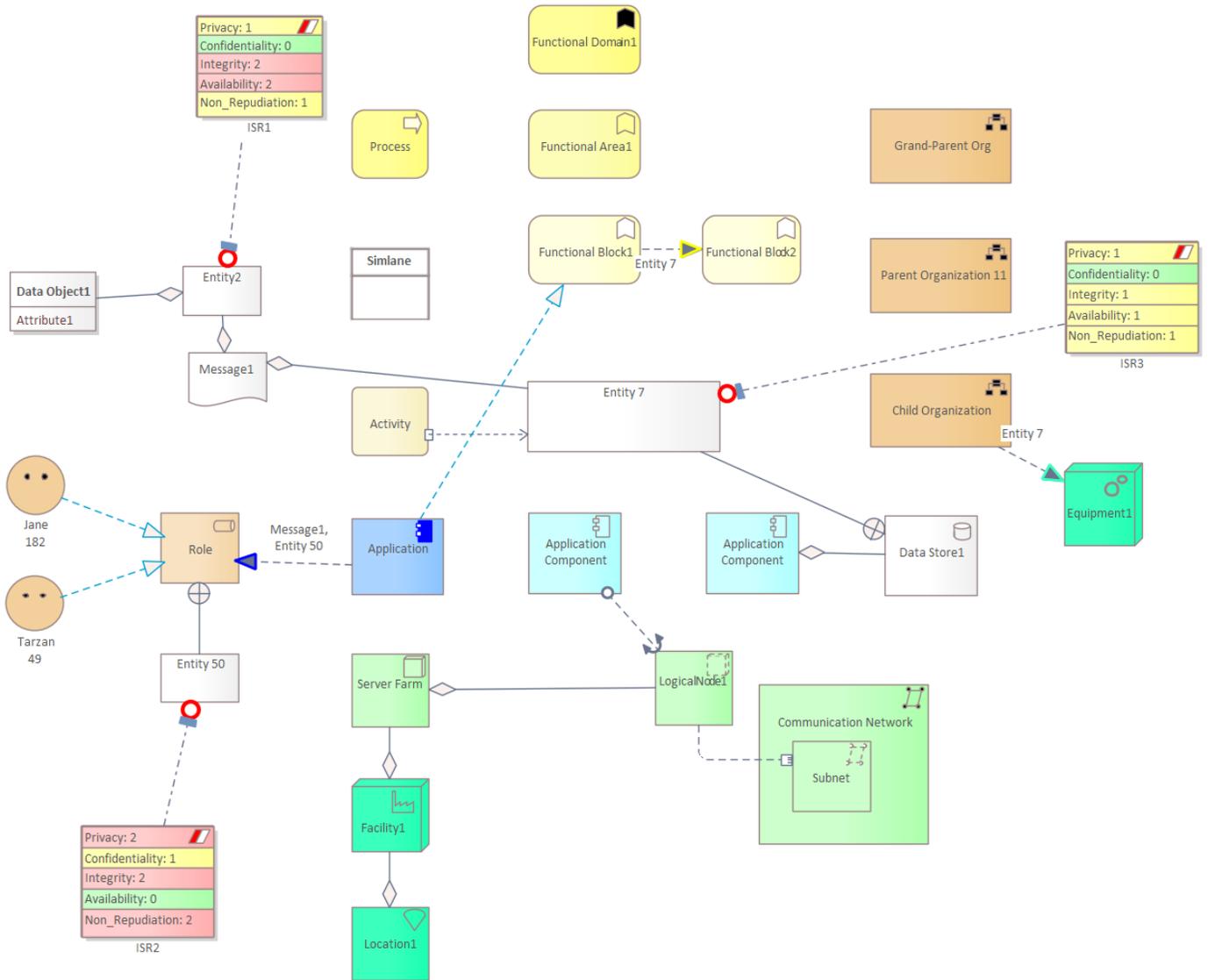
Generate information security requirements for

- Entities
- Entities Exchangers (Processes, Roles, Applications...)

Cancel

Close Help

Sample connectors and information flows that you model



What is automatically generated: implicit connectors and sensitive data usage identification

Based on this generated information, you can provide reports and charts for different audiences.

NB: Information entities are normally not presents in the same diagram as the other items.

The purpose of this diagram is to show what gets generated in terms of connectors and information security properties.

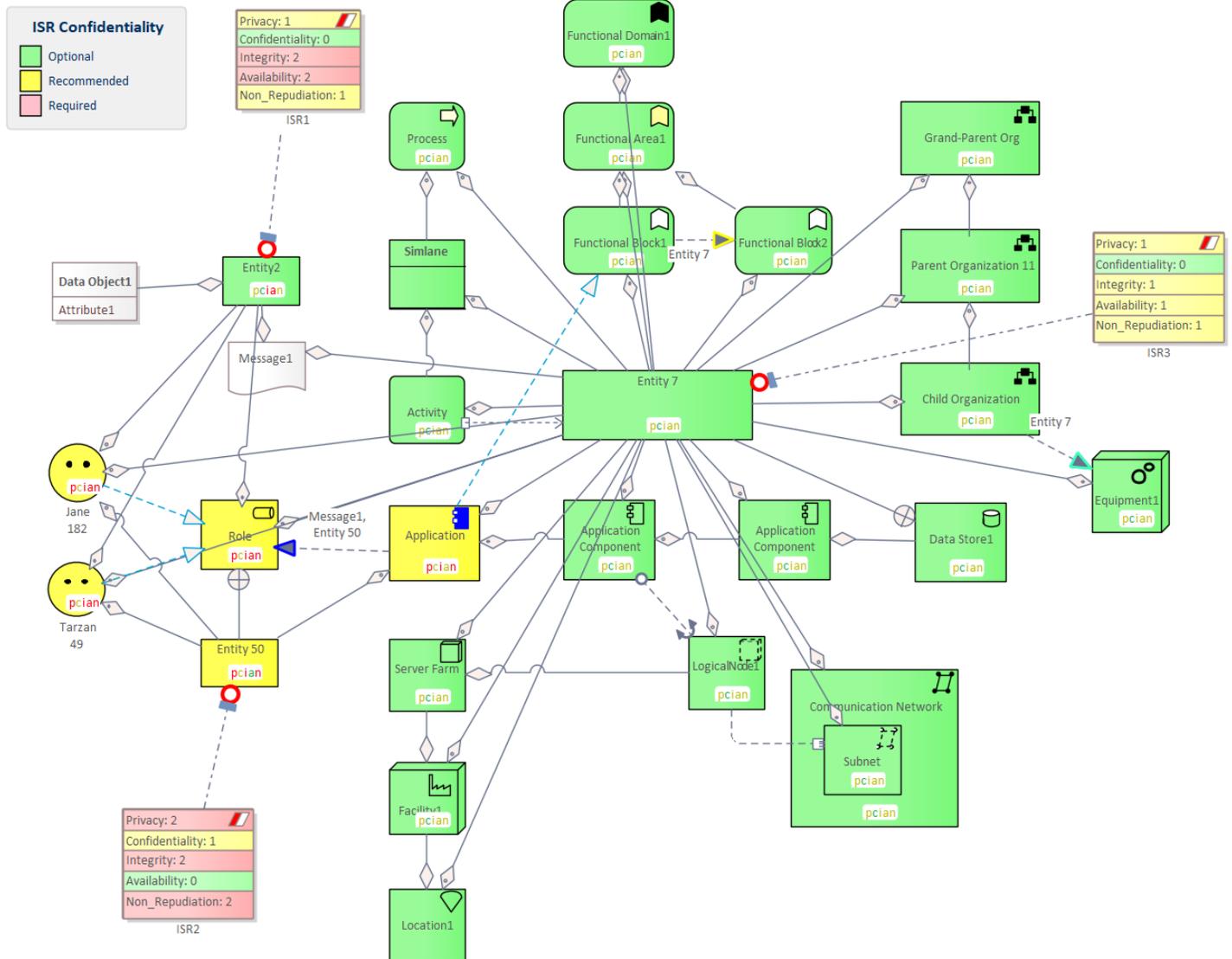
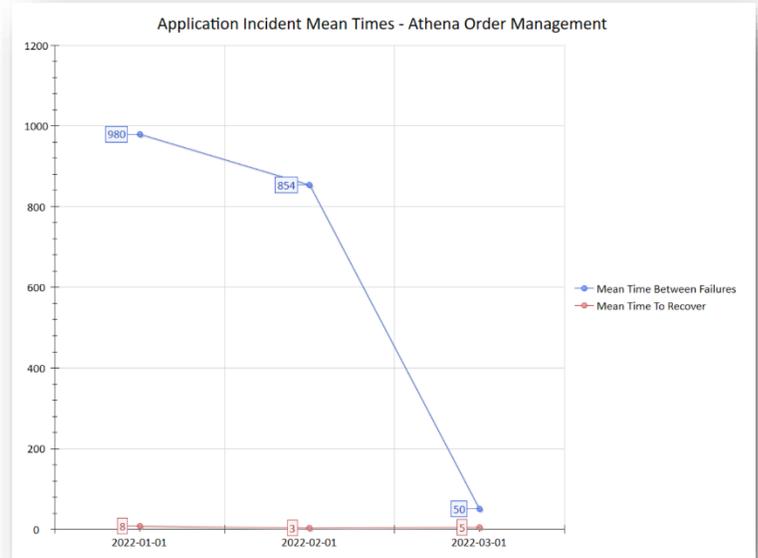
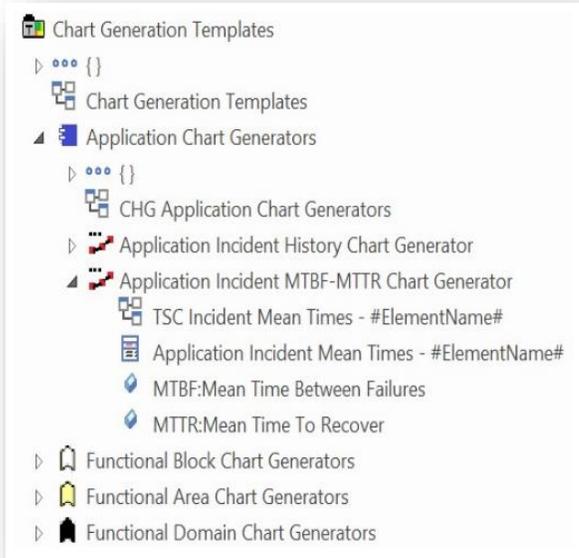


Chart Generation following Templates (Scheduled or On Demand)



- **Non-scheduled chart generators:** If you want a chart generator report to be ignored by the Labnaf PowerShell, then add the prefix "--" to its name.
- [Details on the Labnaf Guidance Web Site.](#)

(Cascaded) Calculations

- **Math expression evaluator**
 - New "=" function that is used for making mathematic calculations between properties of a same element.
 - Support arithmetic operations (+, -, *, /, ^, !, %), Round, RoundDown Abs, Sqrt... and embedded parenthesis.
 - Math Evaluation errors are shown by the Add-in only when the scope is limited to a selected an end user element (for example an application or a process)
- **Calculation on Time Series**
 - Now supports average values calculation across multiple time series
- **Non-scheduled calculations:** If you want a periodical value calculation template to be ignored by the Labnaf PowerShell, then add the prefix "--" to its name.
- [Details on the Labnaf Guidance Web Site](#)

Built-in Metamodel

- Added connections in metamodel
 - Product – Fit -> Product
 - Product – Evolution -> Product
 - Equipment Type – Fit -> Equipment Type
 - Equipment Type – Aggregation -> Equipment Type
 - Equipment Type – Realization -> Equipment Function
 - Product – Aggregation -> Equipment Type
 - Functional Block, Application, Application as a Service, Technology as a Service, Logical Node, Communication Network, System Software – Ownership -> Individual
 - Application Product – Aggregation -> Contract
 - Application, Application Component, Data Store – Dependency -> Application Product

Permissions

- [Generate Diagrams and Charts now requires also “Manage Replicas” permission.](#)

| Labnaf Feature | Required by Sparx EA | | Required by Labnaf | |
|---|----------------------|--------------------------|--------------------------|------------------------------------|
| Calculate Values | 10 Update Elements | | 6 Manage Replicas | |
| Generate Diagram and Chart | 10 Update Elements | 4 Update Diagrams | 6 Manage Replicas | |
| Create Diagram Based On Template Package | 10 Update Elements | 4 Update Diagrams | | |
| Generate Tabular Report | | | 23 Generate Documents | |
| Import Tabular Report | 10 Update Elements | 19 Configure Stereotypes | 12 Import XML | |
| Generate Implicit Data | 10 Update Elements | | 6 Manage Replicas | |
| Merge Elements | 10 Update Elements | 4 Update Diagrams | 6 Manage Replicas | |
| Configure Metamodel - Activate Default Metamodel | 10 Update Elements | | 19 Configure Stereotypes | 38 Configure Project Prerequisites |
| Configure Metamodel - Activate Last Changes In Custom Metamodel | 10 Update Elements | | 19 Configure Stereotypes | 38 Configure Project Prerequisites |
| Configure Metamodel - Set As Custom Metamodel Elements Package | 10 Update Elements | | 19 Configure Stereotypes | 38 Configure Project Prerequisites |

Miscellaneous Labnaf AddIn Improvements

- Adding Legends using Legends form
 - Double click on a legend adds the legend to the active diagram.
 - Simplified legend names

Diagram Generation

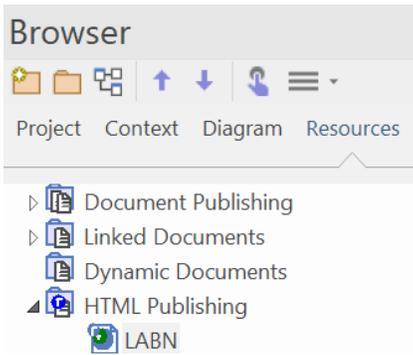
- Merge Diagram generation menu options into one application portfolio generation option that performs following context.

Project Browser Sorting and the Sequence of Operations (Calculation, Chart Generation and Report Generation)

- EA Preferences for the Project Browser are set to "Free Sorting".
Labnaf uses the sequence of elements and attributes in the project browser as the sequence of cascaded calculations. There is no limitation for the sequence of calculations.

Support

- HTML Publication: Labnaf HTML publication template (called “LABN”) for EA 16 (provided in the Repositories folder in your purchased Labnaf package)



- Import: Property values with special characters are now properly imported
- An erroneous calculation expression now results in an empty value (instead of leaving the existing value)
- Generated diagrams: Workaround for Sparx EA 16 color API bug (generated diagram colors are incorrect without the workaround)

Labnaf AddIn

- Add selected legends to one or several diagrams from libraries: Simple legends and Auto Coloring Legends

Legend Libraries

Use the Labnaf menu option "Add Legends To Current Diagram" to automatically add any number of legends to any number of diagrams.

Add your own legends to the legend libraries.

You can store your simple and auto legends libraries in any number of packages.

To make a package part of a Legend library, simply set its stereotype to 'LNCONT_SimpleLegends' or 'LNCONT_AutoLegends'.

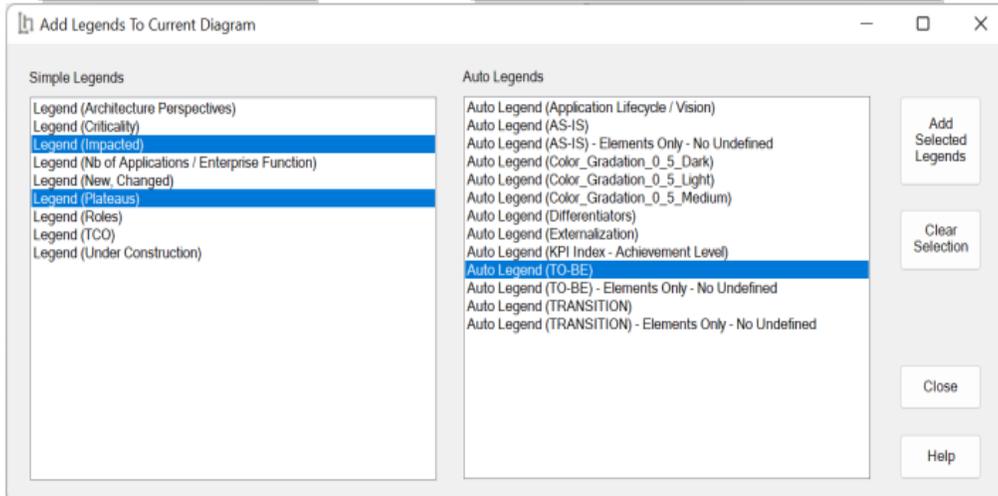
Sparx Systems
Documentation

[Diagram Legends](#)

[Legend Keys](#)

Simple Legends

Automatic Coloring Legends



Automatic coloring legends and diagram stereotypes dynamically change the layout and colors of elements and connectors on your diagrams.

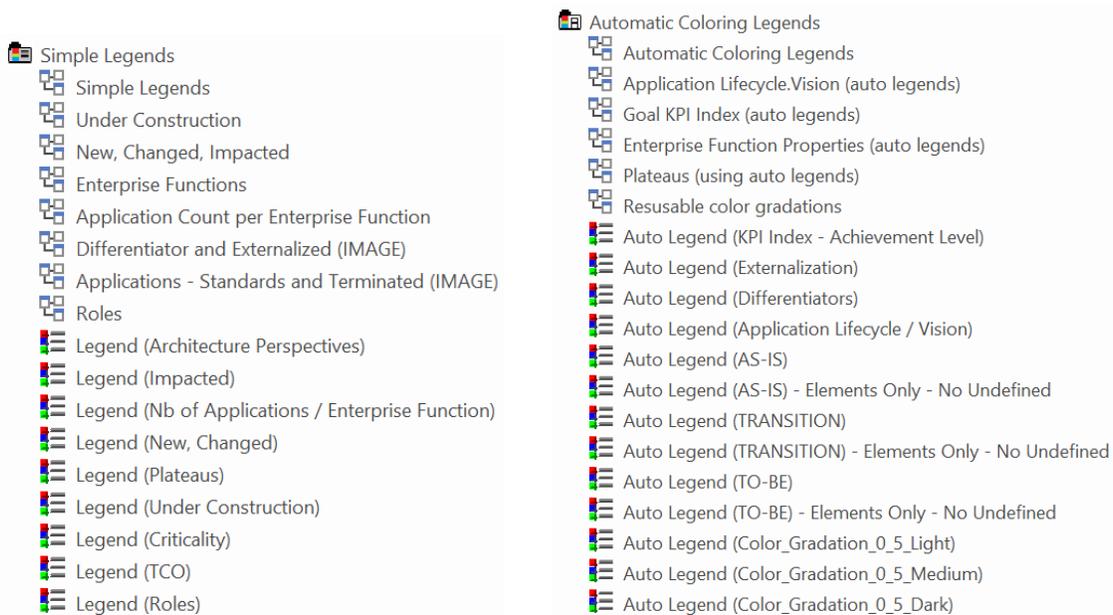
An Automatic Coloring Legend is one that has the 'Auto Color' checkbox ticked.

This puts the Legend into 'active' mode, and will influence how the diagram is actually drawn without you having to make manual changes to the elements on display.

For example, it is possible to set up a Legend that will show all applications that have a specific property with a specific background or outline color or width.

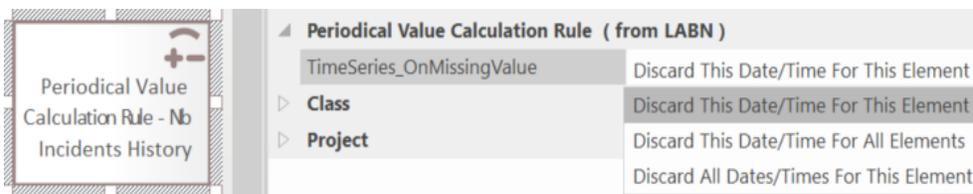
[See also: Alternative layouts for elements and connectors on diagrams](#)

These are the default legend libraries. Each library can span one or several stereotyped packages.



Labnaf AddIn & PowerShell

- Import tabular report (Excel or CSV) ignores header columns. So, if you forgot to remove the generated header columns before importing the report back into the repository then new tagged values will not be created for these header column names. This feature works with Labnaf generated tabular reports created with build B0426 or later.
- Labnaf PowerShell shows start and end date in addition to time
- More detailed error report on explicit scope definition (set of elements to be processed) for calculation, tabular reports, validation and diagram generation.
- New Types of Calculation with [Time Series](#)
 - A time series is a series of values of a quantity obtained at successive times, often with equal intervals between them.
 - Example: History of the number of incidents for each application
 - A Labnaf time series field format has been defined for this purpose
 - Labnaf cascaded calculation engine now enables time series from selected elements to be merged following some rules that can be configured.
 - Example: Applications' history of incidents can be consolidated for functional blocks, areas or domains (which are realized by applications)
- New tagged value for Periodical Value Calculation elements. It is used to define what action is to be taken when a date is missing on a series. ([use the "Synchronize Stereotype" feature to update tagged values in your repository](#))



Labnaf AddIn

Metamodel

The [Labnaf language metamodel](#) can now be dynamically changed directly from the Labnaf AddIn.

This is very convenient both on premises and within the [Enterprise Architect SaaS environment \(cloud\)](#).

A Labnaf metamodel is expressed using the end user language itself. From this metamodel, connector quick linkers and connector validation rules are dynamically generated.

Using the new Labnaf "Configure Metamodel" menu, you can decide to

- either use the default metamodel,
- or customize the default metamodel,
- or create your own metamodel.

To create a new metamodel, you simply use the Configure Metamodel menu option provided by the Labnaf AddIn for Sparx EA to create a "Custom Metamodel Elements Package".

To change the metamodel, you simply add or delete elements and connectors in your custom metamodel elements package.

When satisfied, you activate those changes for all users of the repository.

All in a few clicks.

NB: If you are using the Customization Workbench to customize the MDG, then the Customization Workbench overrides these features, and it is the Customization Workbench that is used to generate connector validation rules and quick linkers from the metamodel.

Built-in Metamodel

- Added connections in metamodel for new Project and Program element types
 - Capability, Feature, Story – Aggregation -> Project
 - Project – Dependency -> Project
 - Program – Dependency -> Program
 - Project – Aggregation -> Program
 - Program – Impact -> Function Domain, Area and Block
 - Project – Impact -> Any resource or function
 - Project and Program – Ownership -> Organization
- Other additions
 - Application Function – Aggregation -> Product
 - Application Platform – Application Flow -> Application Platform
 - Application Group – Application Flow -> Application Group
 - – Ownership -> Organization

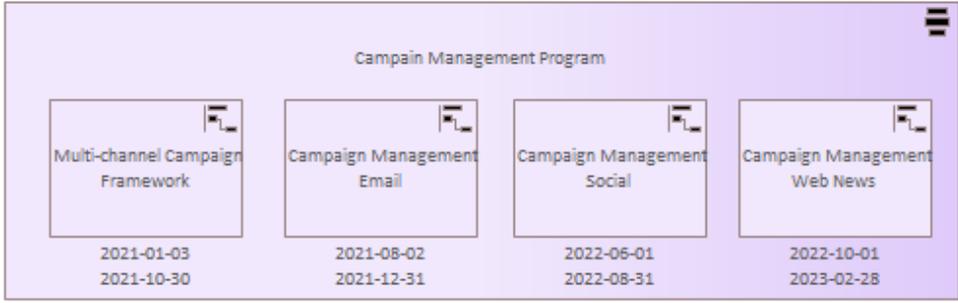
Language

- New element stereotypes: LABN_Project and LABN_Program, which are implementation plans following architecture work
- Added diagram stereotype "Dates Visible" => Start date and end dates can be made visible on diagram for Epics, Projects, Programs, Capabilities, Application Functions, Applications, Equipment, System Software
- Added "Nb_Users" integer type for applications
- Impact connector
 - Added tagged values: Effect = New/Change/Remove => target capabilities, features and epics only impact other elements. They are no longer realized by. Realized by is replaced by Impact with Effect=New.
- Added Viewpoint: Implementation Plan



Diagram stereotype "Dates Visible" is on

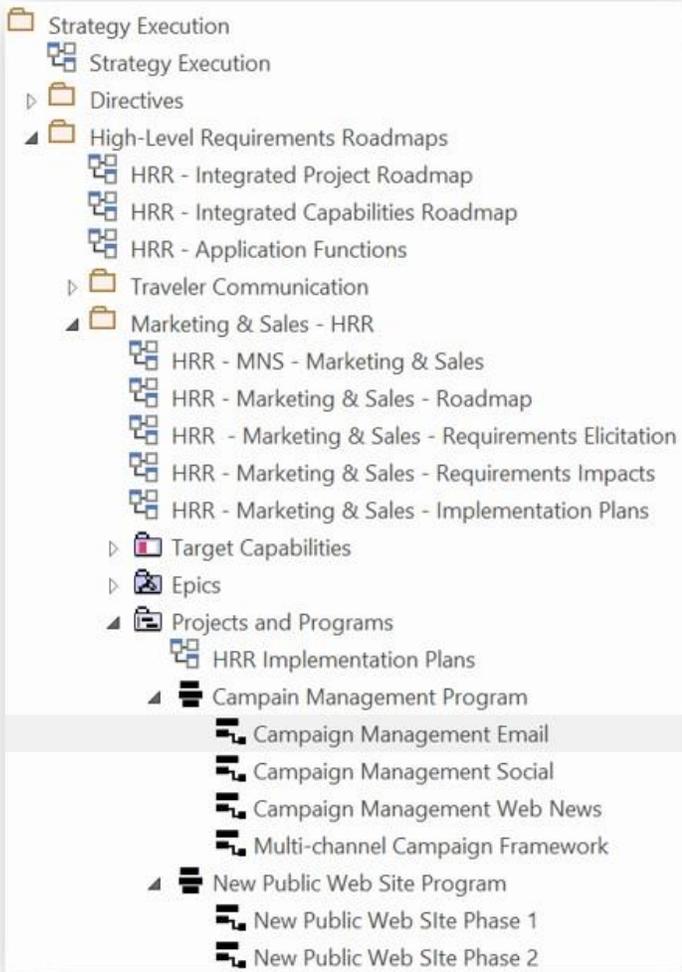
2021-10-20
2023-03-31



2021-01-03
2023-02-28

Repositories

- Sample repository, startup repository and [Canonical repository](#) structure: Added “Projects and Programs” catalog in Strategy Execution



Labnaf AddIn

- Improved performance of menu behavior
- [Search Catalogs](#): Added search on Projects and on Projects that belongs to Programs

The screenshot shows the 'Find in Project' search interface with the following table of results:

| Package_Name | Program_Name | Project_Name | Start_Date | End_Date |
|-----------------------|-----------------------------|----------------------------------|------------|------------|
| Projects and Programs | Campaign Management Program | Campaign Management Email | 2021-08-02 | 2021-12-31 |
| Projects and Programs | Campaign Management Program | Campaign Management Social | 2022-06-01 | 2022-08-31 |
| Projects and Programs | Campaign Management Program | Campaign Management Web News | 2022-10-01 | 2023-02-28 |
| Projects and Programs | Campaign Management Program | Multi-channel Campaign Framework | 2021-01-03 | 2021-10-30 |
| Projects and Programs | New Public Web Site Program | New Public Web Site Phase 1 | 2021-10-20 | 2022-07-30 |
| Projects and Programs | New Public Web Site Program | New Public Web Site Phase 2 | 2022-08-01 | 2023-03-31 |

Labnaf PowerShell

- Validation rule configuration accepts the case where the structure of enterprise functions starts directly with a functional area (no functional domain)

Implicit Connector Generation

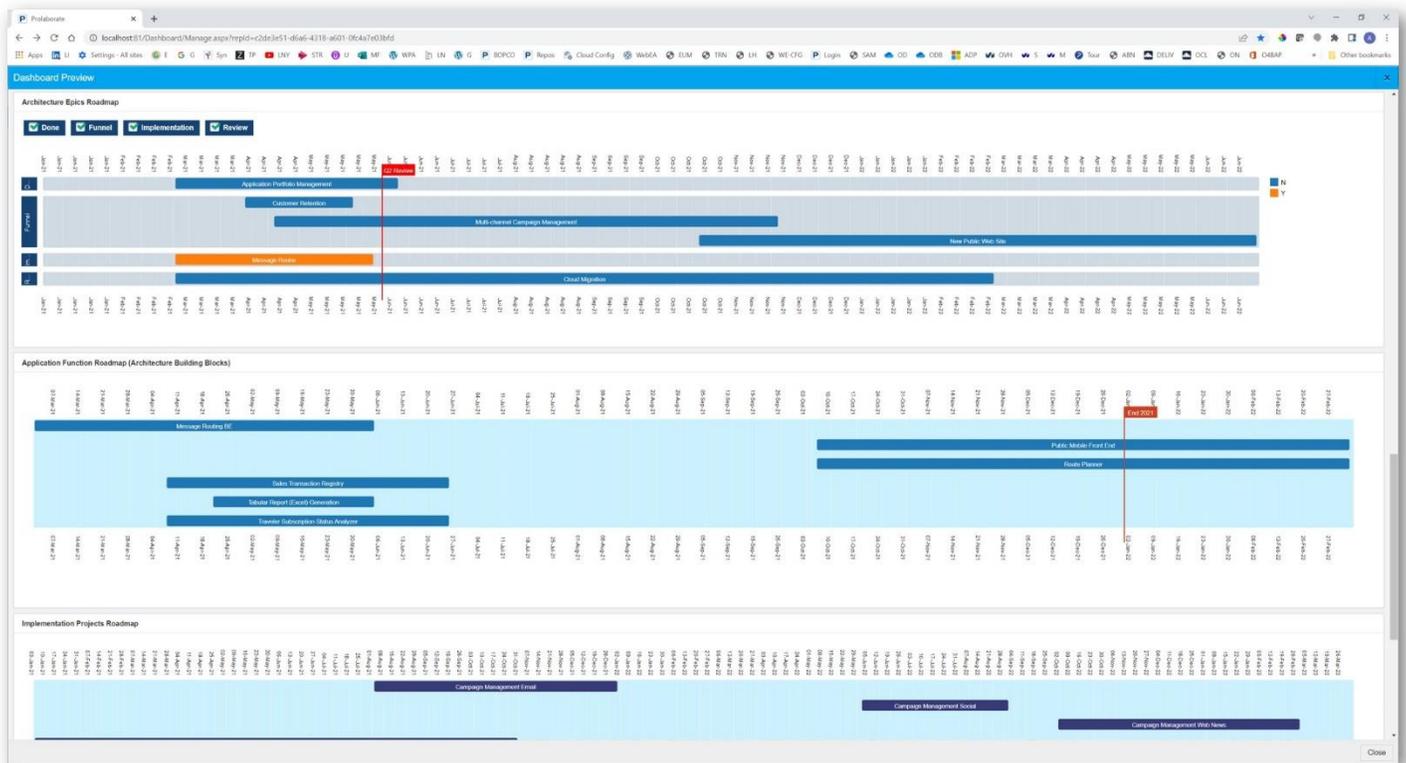
- New implicit aggregation generation for hierarchy of Projects in Programs

Calculation and Tabular Report Generation

- Add cascaded calculation of incidents for applications, functional block, area and domain
- The sequence of calculations and reports is now following the sequence visible in the browser window.

Charts

- Added Prolaborate charts for Application Function (ABB) and Projects roadmaps



B0423

Labnaf AddIn & Labnaf PowerShell

- Different users can use different date formats. Dates manually entered or imported with any user specific format are automatically normalized to YYYY-MM-DD in the database so that Prolaborate roadmaps can be properly displayed.

Labnaf PowerShell

- Commands are more verbose.

Language

- Functional Domain, Functional Area, and Functional Block
 - Added tagged values: Nb_Incidents (cascaded calculations based on Application.Nb_Incidents)
- Application
 - Added tagged values: Nb_Incidents
- Application Function

- Added tagged values: Impl_Start_Date & Impl_End_Date. A new calculations automatically calculates these dates based on the dates of the target capabilities that impact those application functions.

(use the “Synchronize Stereotype” feature to update tagged values in your repository)

Labnaf AddIn

- [Search Catalogs](#): Added search on Applications Realizing Enterprise Functions (Functional Domains, Functional Areas, and Functional blocks)
- No more error message when using an empty repository (without any Labnaf configuration).

Repository

Calculations

- Cascaded calculation of the number of incidents per functional block, area, and domain based on the number of incidents per application.

B0422

Labnaf AddIn and Labnaf PowerShell

- Fixed diagram generation cleanup. One element was not properly deleted during re-generation of existing diagram.

B0421

Labnaf AddIn and Labnaf PowerShell

- [Implicit Connector Generation](#) (including 360° view on information usage)
 - Updating implicit connectors now eliminates all generated connectors that should no longer exist
 - Improved progress tracking in the Labnaf PowerShell

Startup & Sample Repositories

- Added sample [Excel report template to generate 360° view on information usage](#).

| Information Type | 1. Information Use | 2. Information Use | 3. Information Use | 4. Information Use |
|--------------------------------|--------------------|--------------------|--------------------|--------------------|
| Account | | | | |
| Application Architecture Data | | | | |
| Application Configuration Data | | | | |
| Applications | | | | |
| BOPCQ Identity | | | | |
| Bank Marketing and In-S | | | | |
| Customer | | | | |
| Customer | | | | |
| Customer Center | | | | |
| Customer Product | | | | |
| Geography | | | | |
| Party | | | | |
| Person | | | | |
| Person | | | | |
| Person Location | | | | |
| Product | | | | |
| Product Price | | | | |
| Product | | | | |
| Product Location | | | | |
| Resource Location | | | | |
| Route | | | | |
| Customer-Product | | | | |
| Supplier-Product | | | | |
| Vehicle Location | | | | |

Labnaf AddIn and Labnaf PowerShell

- [Implicit Connector Generation](#)

Generate implicit connectors following numerous patterns and options, including:

- for child elements following defined element stereotype hierarchies (this option is enabled by default)
- for information elements (entities) used, owned, aggregated or exchanged directly or indirectly by any other element.

BENEFITS OF IMPLICIT CONNECTOR GENERATION

- **360° view on information usage.** Know which information is used by whom, by which role, organization, function, process, activity, data flow, application, component, data store, server, equipment, network, etc
- Know which information is stored where (for example in which country)
- Dramatically simplifies information security and **GDPR** compliance analysis
- Simplifies traceability as embedded elements get aggregation connectors
- Normalizes the way elements are related in a repository i.e. based on connectors
- Enables powerful reporting capabilities including Prolaborate charts

Implicit connector generation identifies and removes previously generated connector that are no longer relevant. So, there is no need to delete connectors before generation.

[More information about the various implicit connector options on the Labnaf Guidance Web Site.](#)

Labnaf AddIn

- [Merge Elements and Connectors](#) (Merge Redundant Items)

As a modeler, you often meet situations where you need to merge redundant items. Redundancy typically happens when several teams created copies of the same items.

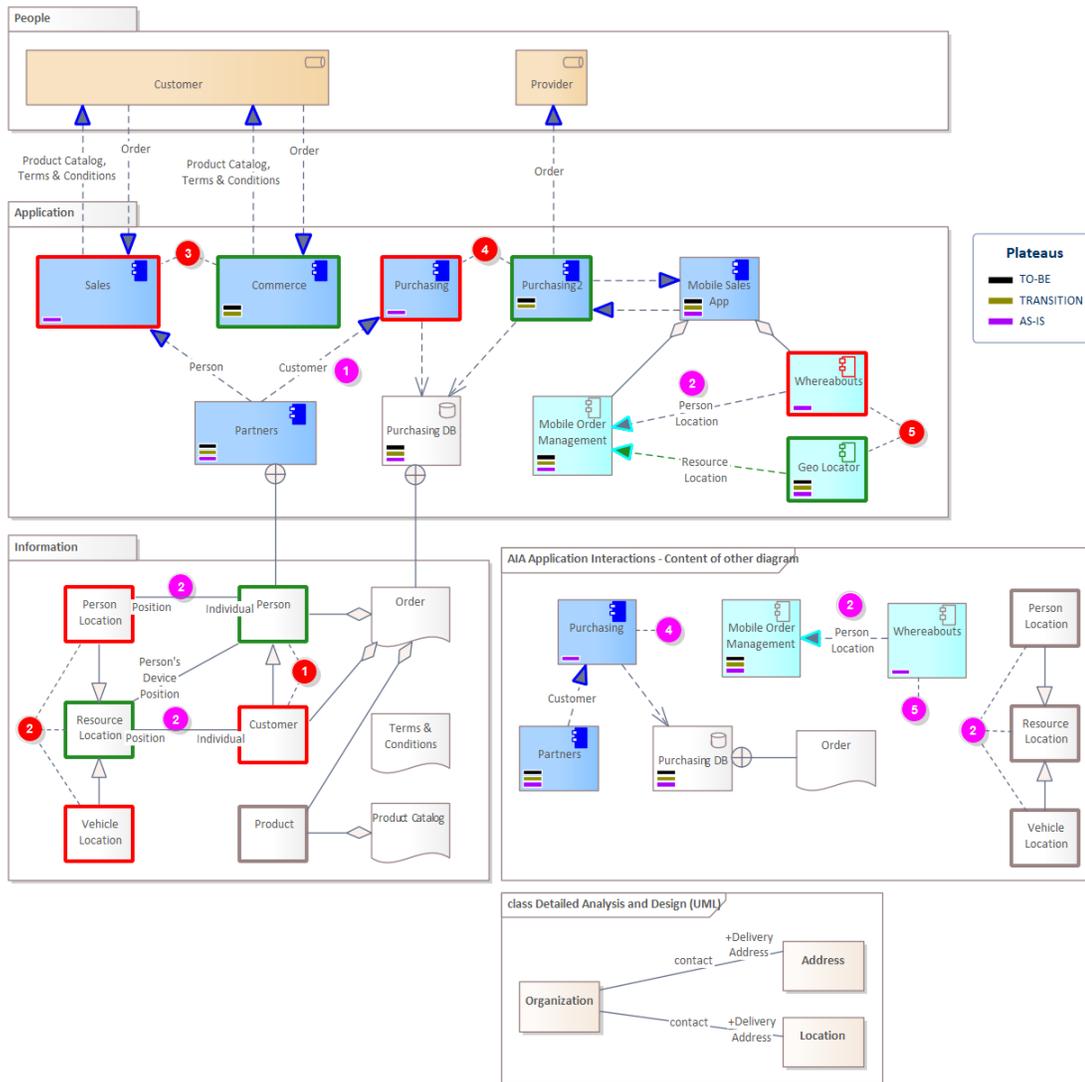
Labnaf can merge any number of duplicate elements into one element to be preserved.

During that element merge process, Labnaf also

- collects impacted connectors and merges them if applicable,
- updates conveyed object on impacted information flows,
- moves child elements and diagrams from duplicate elements to preserved element, and
- updates all impacted diagrams.

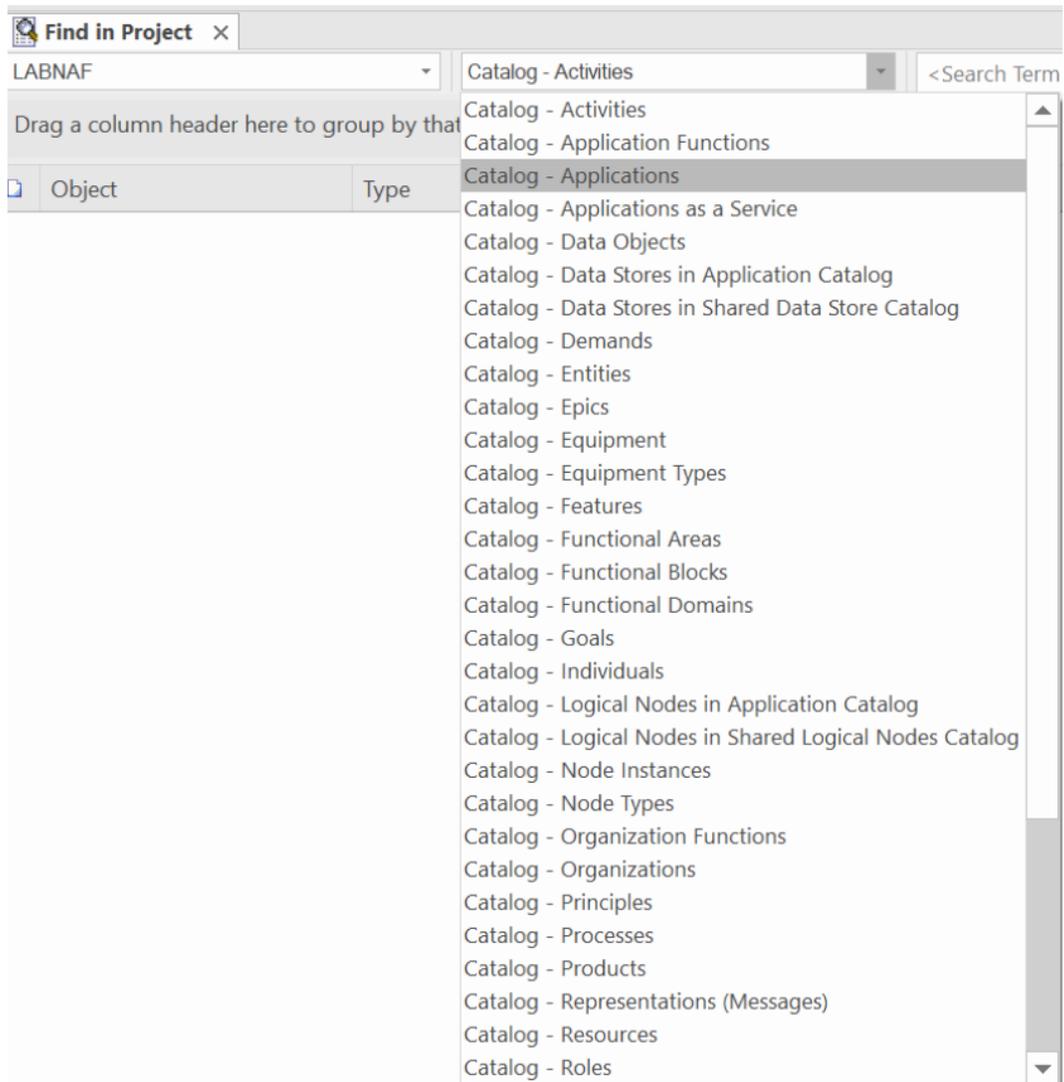
The following sample diagram shows

- a sequence of numbered element merge actions, and
- the redundant elements to be merged into one element to be preserved
- the side effect of each merge action.



[More information on the Labnaf Guidance web site.](#)

- Conveyed items on functional flows, application flows and component flows can now be either representations, entities or data objects.
- [Search Catalogs](#)
 Each catalog search returns all elements that belong to a specific Labnaf 'virtual' catalog. These elements can be distributed in any number of Labnaf 'catalog packages' throughout the repository.



[More information on the Labnaf Guidance Web Site.](#)

Security

- [Permissions](#)

Access to different Labnaf functionalities is now controlled using users' permissions.

Labnaf fine management of permissions is now based on Sparx permissions. The Sparx built-in user interface for configuring users and groups permissions is also used for configuring Labnaf permissions.

Permissions can be assigned to entire groups and/or to specific users. Dedicated users' permissions add to the permissions of the groups that they belong to.

[More information on the Labnaf Guidance Web Site.](#)

- [Locking mechanisms](#)

Trying to merge locked elements

Labnaf will not let users merge any element in the following cases

- at least one of the elements to be merged is locked
 - for the current user
 - for one of the groups that the user belongs to
- a parent package of one of the elements is locked
- the user does not have the required permissions to merge elements

Trying to import Changes to Existing elements

The Labnaf Addin raises an error and halts the import process.

The Labnaf PowerShell raises an error, but process continues with the next element in scope.

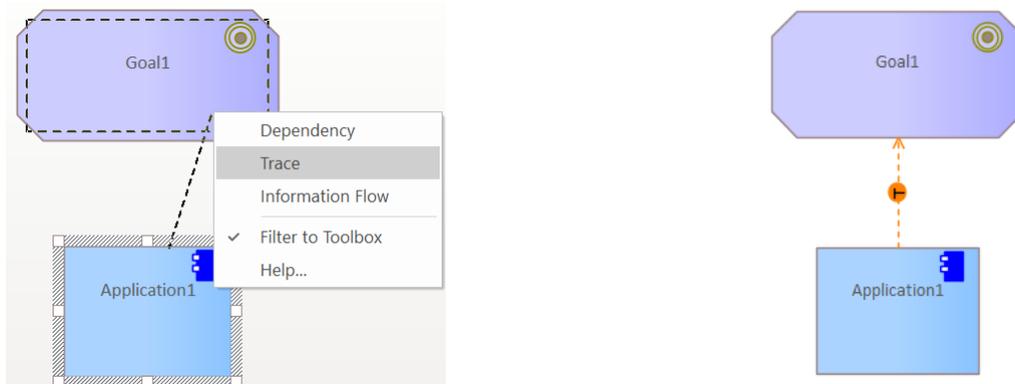
Trying to import New elements to a locked package

The Labnaf Addin and the Labnaf PowerShell raise an error and halts the import process.

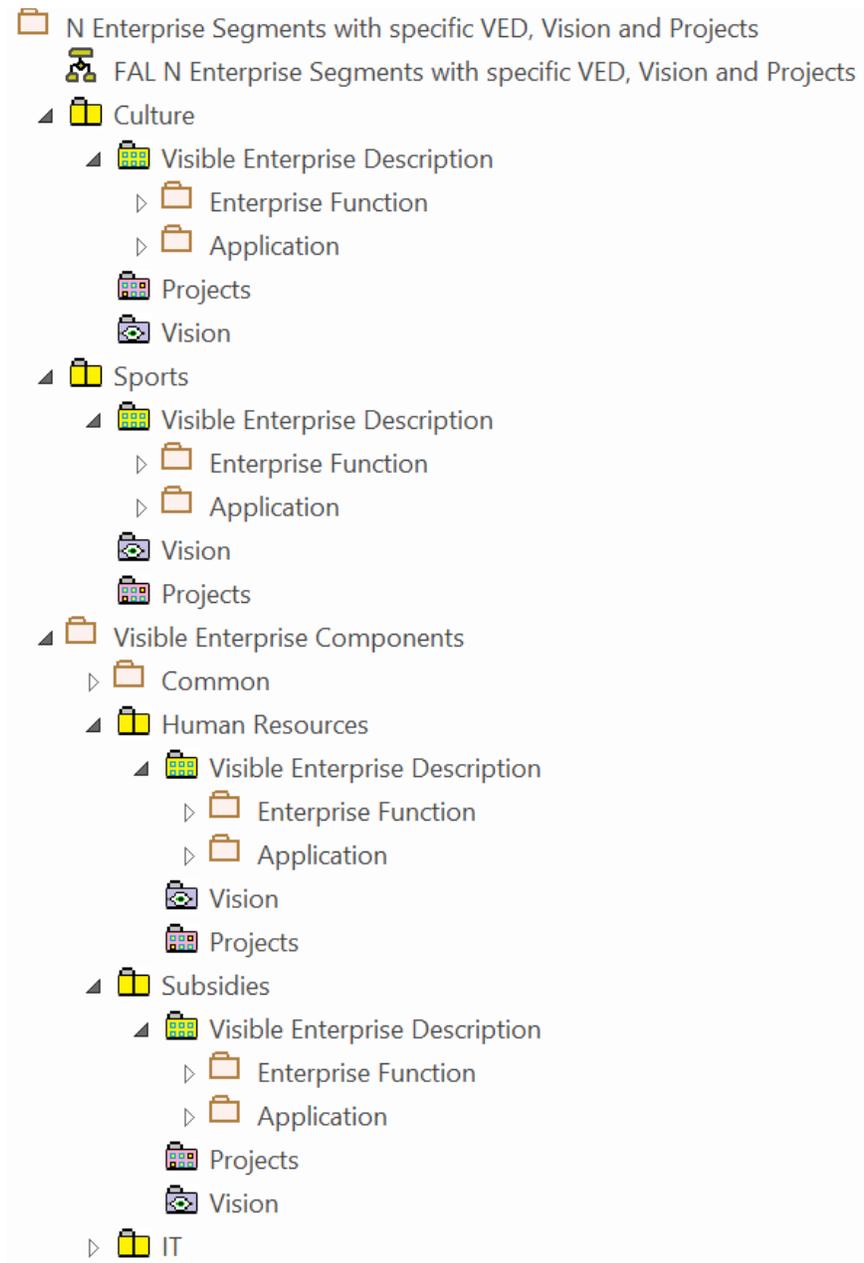
Language

- Added connector type

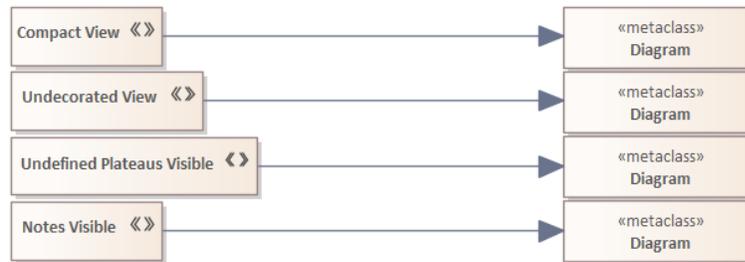
- **Temporary Trace:** Can be created to temporarily connect elements of types that are not meant to be connected.
- To add a temporary trace connector, use the generic UML "Trace" connector. The connector shows as below.



- Added Package stereotype LNCONT_EnterpriseSegment
 - Related existing package stereotypes: LNCONT_VisibleEnterprise, LNCONT_Vision, LNCONT_ProjectDeliverables
 - These stereotypes can be used to model complex organizations with multiple segments. Leverages virtual catalogs that can be made of any number of visual enterprises and catalog packages.



- Added alternative layouts for elements and connectors and diagrams.
Use the [diagram stereotype values](#) to change the layout of elements or connectors in diagrams. The diagram stereotype can be set using the diagram Properties => Stereotypes drop-down list.
- Added related diagram stereotypes



- Node instance
 - Added tagged value
(use the [“Synchronize Stereotype” feature to update tagged values in your repository](#))
 - IP_Address that shows on the shape itself

Metamodel

- Added connections in metamodel so that processes and activities can access entities (in addition to representations)
 - Process & Activity <- Access -> Entity
 - Application Flow & Component Flow can convey Entity, Data Object (in addition to Representation)
 - All performer element, functional performer, process, activity, node and network elements can aggregate entities. This will be leveraged in an upcoming version of Labnaf for sensitive information and risk management.
 - All performers can specialize performers of the same type
 - Capability & Epic – Ownership -> Organization
- Added connections in metamodel (for consistency with connector generation)
 - Activity, Swim-lane, Application as a Service <- Aggregation -- Entity
- Bug fix: Fixed Specialization connector quick linkers verbs (displayed when creating a new connector).

Startup Repository

- [New “DEFAULT Language Metamodel - My Changes”](#)
 - Use this diagram to
 - model the connections that you added to the default language metamodel (model)
 - document the connectors that you removed from the default language metamodel (textual description)

B0415

Labnaf Addin

Language

- Added connections in metamodel
 - Entity – *Is Part Of* -> Functional Domain / Area / Block / Category / Service

Labnaf PowerShell

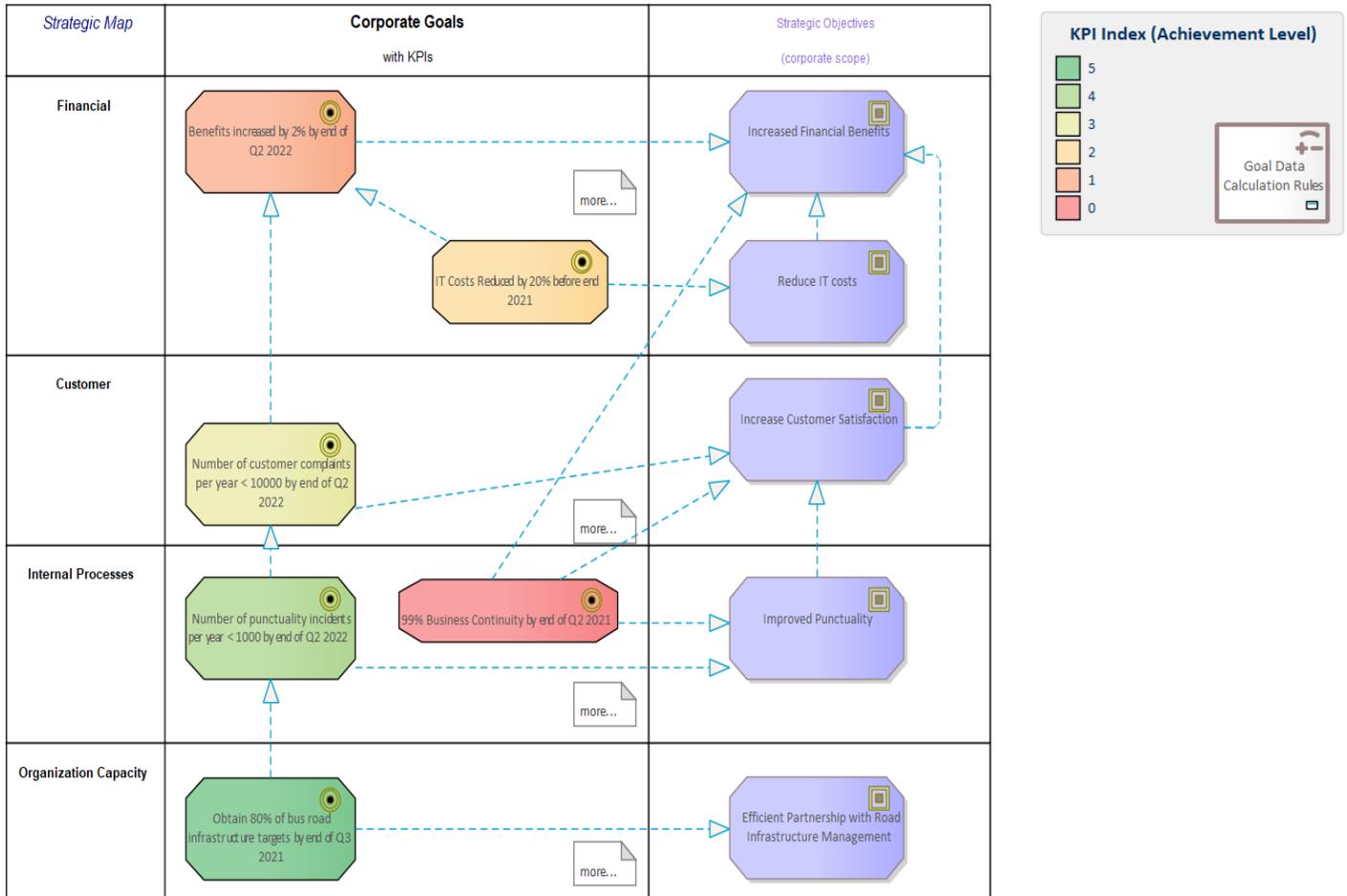
- ImportConnectors
 - Corrected problem with connector stereotype creation
- MoveElementsToCalculatedParent
 - Corrected reversed ParentToChild / ChildToParent connector direction handling
 - Added 'Verbose' option to display the name of each element being moved along with the name of its new parent.

Labnaf Addin

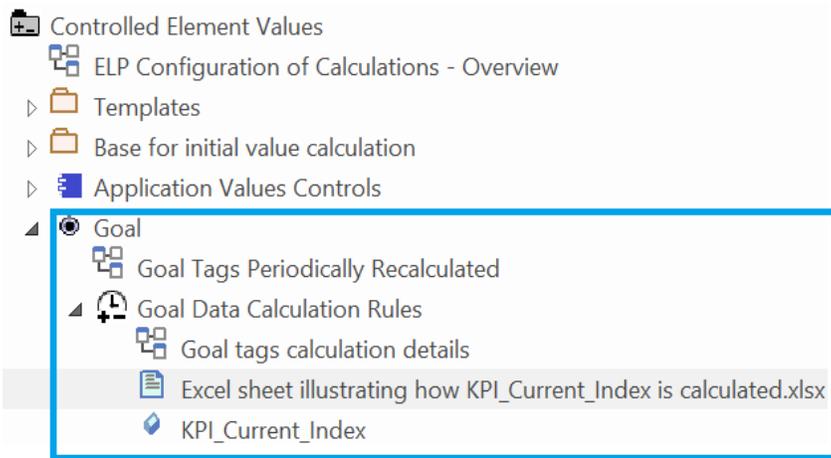
- Any change to any value calculation rule has now immediate effect (useful for testing).

Language

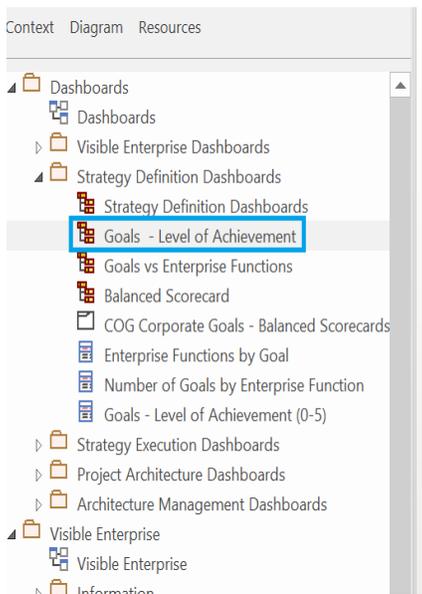
- Goal
 - Added tagged values and calculation to govern goal achievement (use the "Synchronize Stereotype" feature to update tagged values in your repository).
 - KPI_Current_Index (calculated based on items highlighted in yellow).
 - Measure_Of_Success
 - KPI_Target
 - KPI_Current
 - KPI_Baseline
 - Goal_Status
 - Start_Date
 - End_Date
 - Related new tagged value types: KPI_Target, KPI_Current, KPI_Baseline, Goal_Status



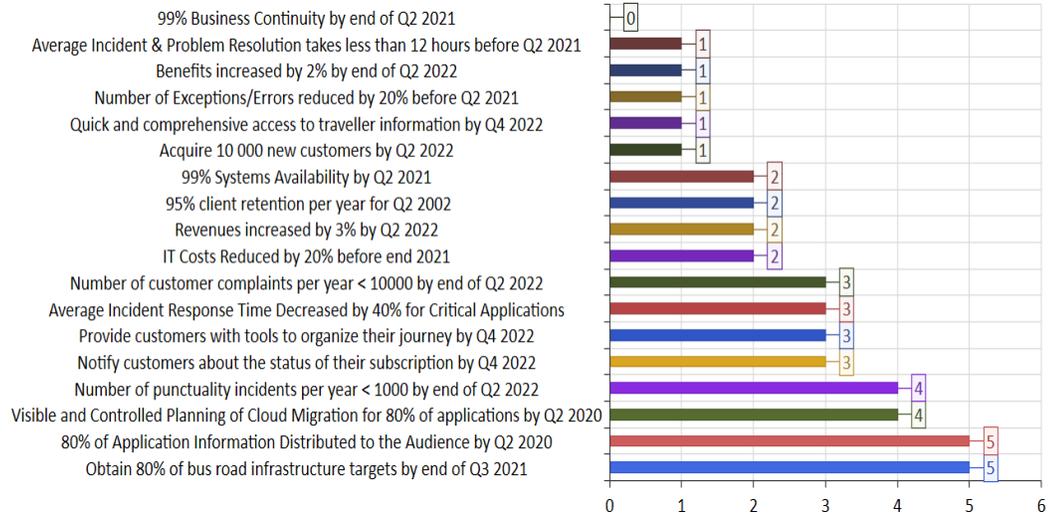
- Added Calculations for goal's KPI_Current_Index



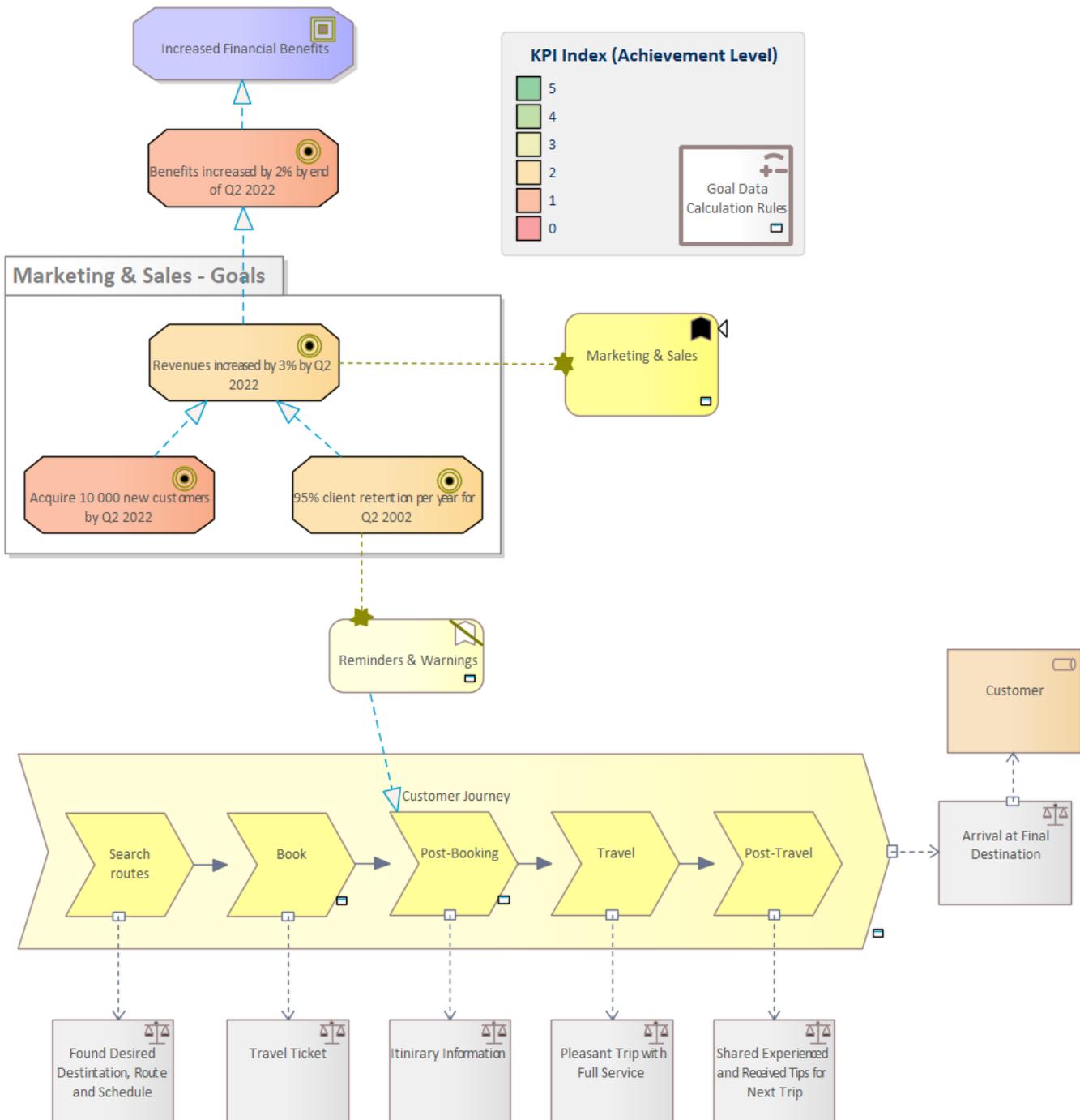
- Added sample chart to govern the SMART goals' level of achievement



Goals - Level of Achievement (0-5)



- In the sample repository, all SMART goals and related sample diagram now have a calculated KPI Index and are automatically colored.



- Added sample diagram with the list of corporate goals not aligned to any strategic objective

ext Diagram Resources

- ▲ Strategy Definition
 - Strategy Definition
 - Context Analysis & Diagnosis
 - Corporate Strategic Foundations
 - ▲ Strategic Plans
 - Strategic Plans
 - Corporate Strategic Themes
 - Corporate Strategic Objectives
 - ▲ Corporate Goals
 - COG Corporate Goals - Balanced Scorecards
 - List of Corporate Goals not aligned to any Strategic Objective
 - Benefits increased by 2% by end of Q2 2022
 - Number of customer complaints per year < 10000 by end of Q2 2022
 - Number of punctuality incidents per year < 1000 by end of Q2 2022
 - Obtain 80% of bus road infrastructure targets by end of Q3 2021
 - Quick and comprehensive access to traveller information by Q4 2022
 - IT Costs Reduced by 20% before end 2021
 - List of Corporate Goals not aligned to any Strategic Objective
 - Domain Goals
 - CSM - Strategic Corporate objectives 2021
 - Strategic Plans

- Strategy Execution
- BMC Business Model
- CSF Company Values
- CSF Strategic Foundations
- VPC Value Proposition for Commuters
- Projects
- Projects
- ▲ Subscription Expiry Alerts
- Subscription Expiry Alerts

List of Corporate Goals not aligned to any Strategic Objective

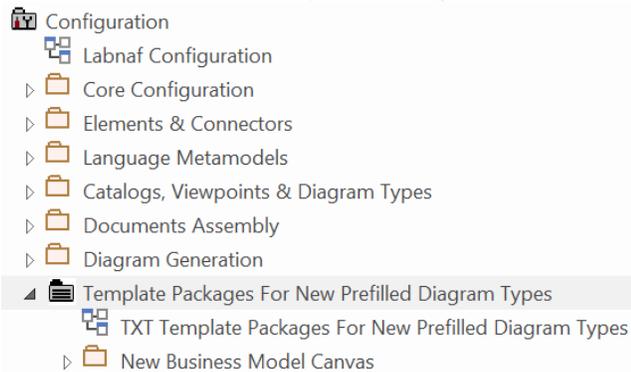
Corporate Goals

- 95% client retention per year for Q2 2002
- 80% of Application Information Distributed to the Audience by Q2 2020
- 99% Systems Availability by Q2 2021
- Acquire 10 000 new customers by Q2 2022
- Architecture Visibility, Traceability, Reuse & Consistency
- Architecture-driven Business Transformation
- Average Incident & Problem Resolution takes less than 12 hours before Q2 2021
- Average Incident Response Time Decreased by 40% for Critical Applications
- Better System Design, Sizing & Configuration
- Centralized Dashboards & Reports
- End-user Experience & Application Monitoring
- Increased Visibility on Service Structure & Quality
- Integrated IT Architecture and Operation Informaton
- IT Landscape Rationalization
- Notify customers about the status of their subscription by Q4 2022
- Number of Exceptions/Errors reduced by 20% before Q2 2021
- Outsourced management of 80% of applications and servers by Q3 2021
- Provide customers with tools to organize their journey by Q4 2022
- Revenues increased by 3% by Q2 2022
- Visible and Controlled Planning of Cloud Migration for 80% of applications by Q2 2020

B0413

Language

- Added connections in metamodel
 - Node instance – Node to network connection → Communication Network
 - Role – Realization → Functional Block and Functional Area
 - [Logical Node \(Firewall\) – Flow Allowed By Firewall → Application Flow or Component Flow \(connector\)](#)
- Instance Deployment and Connectivity toolboxes
 - Add node instance and connector to communication network
- The existence of the « Template Packages For New Prefilled Diagram Types» is no longer mandatory.



B0412

Language

- Added connections in metamodel for the [connectors that can be automatically generated](#) following the element hierarchy in the browser window. So that these connectors can be also created manually.
 - Data Object – *Is Part Of* -> Entity
 - Location – *Is Part Of* -> Location
 - Facility – *Is Part Of* -> Facility
 - Capability – *Is Part Of* -> Capability
 - Feature – *Is Part Of* -> Feature
 - Functional Service – *Is Part Of* -> Functional Category – *Is Part Of* -> Functional Block – *Is Part Of* -> Functional Area – *Is Part Of* -> Functional Domain
 - Equipment Function – *Is Part Of* -> Equipment Function
 - Organization Function – *Is Part Of* -> Organization Function
 - Technology Function – *Is Part Of* -> Technology Function

Guidance

- Added the list of [patterns of element type hierarchies](#) for which connectors can be generated.

B0411

Labnaf Addin

New Labnaf menu option: “Open Contextual Guidance”

Opens guidance web site. [If the current diagram is a Labnaf diagram, then the guidance web site opens on the related viewpoint/diagram type documentation.](#)

New diagrams can be prefilled following user’s configuration

For a few diagram types, like the business model canvas, it is easier when the [content is automatically prefilled, for example with swim lanes, auto-coloring legends, default elements etc.](#) Since many elements can be automatically created, we don’t want to mix the automatically created elements with our existing elements. So, it is better to store these automatically in a new separate package, even if that package is temporary.

Guidance and HTML Generation

- [Prolaborate charts and dashboards \(documentation provided as a video\)](#)
- Generated HTML for text and notes items is no longer clickable, as this was resulting to a dead end. This applies also to the on-line guidance.
- New legends and automatic coloring legends

Language

- **[Business Model Canvas](#)**
 - New diagram type
 - By default, new Business Model Canvas diagrams are automatically created in a new package containing all swim lanes and auto-coloring legends needed. This is using a new feature called “template package for prefilled diagrams”.
 - New element types
 - Cost Type
 - Revenue Stream
 - Customer Relationship Type
 - Besides the above new element types, existing element types are reused for the Business Model Canvas
- **[Value Proposition Canvas](#)**
 - New diagram type
 - New element types
 - Customer Job
 - Desired Gain
 - Pain
 - Besides the above new element types, existing element types are reused for the Value Proposition Canvas
 - New connector type
 - Fit
- **Value** element type extensions for its usage in the Value Proposition Canvas
 - Added tagged values ([use the “Synchronize Stereotype” feature to update tagged values in your repository](#))
 - Differentiator
 - Is_Gain_Creator
 - Is_Pain_Reliever

- Relevance
- **Functional Domain, Functional Area, and Functional Block**
 - Default value for Differentiator is now 'N'
- The [conceptual metamodel](#) and the [language metamodel](#) integrate the new concepts and elements added for the business model canvas and the value proposition canvas.

Repositories

- Enriched startup repository and sample repository
 - [Much improved startup page for new Labnaf users](#)
 - Business Model Canvas and Value Proposition Canvas
 - Configuration of template packages for prefilled diagram types (used for Business Model Canvas as an example)

B0402

Language

- Changed "Functional Domain", "Functional Area", "Functional Block" element types
 - Added tagged values ([use the "Synchronize Stereotype" feature to update tagged values in your repository](#))
 - Efficiency (1-5)
 - Effectiveness (1-5)
 - Value_Type (Financial, Customer, Other)
- Added connections in metamodel
 - Functional Domain ← *Functional Flow* → Role

Labnaf PowerShell and Language Transformer

- Provides information about license status (can be turned off)
- Return error code (0/1) in addition to description

Labnaf PowerShell

- Validation command
 - provides information about the process of the ongoing validation.
 - detailed warning when more than one recipient/individual with same email address

Sample repository and Guidance

- Added examples for enterprise function effectiveness and efficiency. Useful for Prolaborate dashboards.

License Manager

- Shows version number and build number.

B0400

Labnaf Addin

Extended Import GUID

- Add new elements and/or update existing elements
- Match external and internal elements using

- a unique key (EA_GUID, a tagged value, the name or the alias)
- an option external vs internal property name mapping table
- try alternative unique keys in a defined sequence

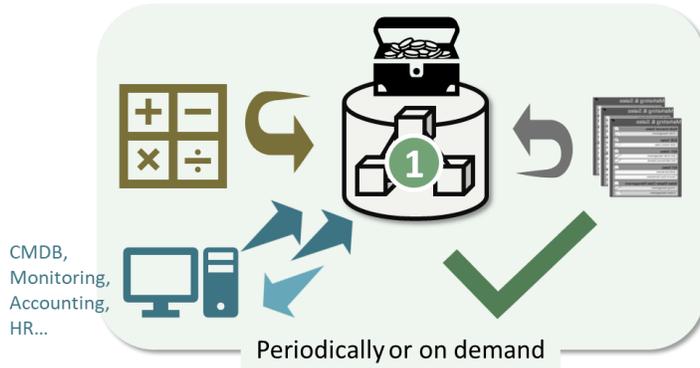
Labnaf PowerShell

- The Labnaf PowerShell has been enriched with many commands. These new commands can be used to create sophisticated systems integration mechanisms, model refactoring and automatic structure normalization.

Data Enrichment and Distribution for Enterprise Visualization

Labnaf enriches your data

- Imports data from different sources
- Cascades calculations
- Generates diagrams
- Normalizes the repository structure
- Exports to other systems and repositories



Sparx EA & Prolaborate present the data

All models and charts are reusable. Because there is one single language, metamodel and repository structure



- New commands:
 - ClonePackage
 - CreatePackage
 - ExportToXmi
 - ImportConnections
 - ImportFromXmi
 - MoveElementsToCalculatedParent
 - MoveElementsToPackage
 - MovePackagesToPackage
 - RenameItem
 - ScheduleCommand
 - SetDiagramProperty
 - SqlExportToCsv

Extended command

- ImportTabularReport: Ability to import new elements or update existing elements using one or several unique keys (EA_GUID, a tagged value, the name or the alias). Try alternative unique keys in a defined sequence
- Numerous samples scripts to address different integration situations.

Labnaf Language Transformer

- Improved element transformation.
- Default child diagram setting for the transformed element

B0322

Labnaf Addin

- When a new diagram is created, a three letters prefix is added to the diagram name to indicate the type of diagram/viewpoint ([see the three letters acronym defined for each viewpoint](#))

Language

- Added diagram type variants that include swim lanes
 - Process (SIPOC)
 - Strategy Map
- Added catalog of features (when working with features only, without parent capability)
- Added connections in metamodel
 - Value realizes Goal
 - Node Type aggregates Equipment Type
 - Logical Node aggregates Equipment

Viewpoints

- In High-Level Requirements Roadmap / Goal-Driven Requirements Elicitation, added approach for identifying target capabilities from value streams. Either via the values provided by the value streams, or via the enterprise functions that are orchestrated by value streams.

Sample repository

- Added examples to illustrate robotics, IoT, ecosystems, SIPOCs and cascaded calculations used in heat maps.

Productivity Tools

Extended Auto-Connect feature for generating aggregations for all child elements following defined embedding rules.

B0321

Language

- Extended support for Value, Value Proposition, Product and Product Line
- Product can be Goods, Services or Mix
- Application, System Software and Equipment
 - Added tagged value In_Operation_Start_Date and In_Operation_End_Date (for portfolio management and roadmaps)
- Added connections in metamodel
 - Connectors for Value, Value Proposition, Product and Product Line
 - Application Function – *Realization* -> Swimlane

Sample repository

- Added examples for value streams, product lines and value proposition

Charts & Value Calculation

- Sample cascaded calculations and chart for functional taxonomy (aka business capability map)

B0319

Language

- Added “Value” element type + related metamodel connections.
- Grouped “Resource” and “Value” into “Generic” set of elements (in toolboxes)

Labnaf Addin

- On demand calculation – Testing mode: Calculate tagged value for selected user element and for all applicable calculation rules.

Repositories

- Sample repository: Sample 3 levels cascaded calculation + usage of results in enterprise function/capability heat map.

B0318

Language

- Added tagged value for Technology as a Service
 - Origin = External (default) or Internal. Each value translates to a specific element shape.

Labnaf Addin Lite

- Can now be used to open SQL Server repositories in read only mode

B0317

Productivity Tools

- New Auto-Connect feature for generating aggregations for all child elements following defined embedding rules. This is important for creating **Prolaborate** charts as these rely on relationships and not on element embedding. This feature can be used either on demand or it can be scheduled using the Labnaf PowerShell.
- Validation configuration is much simplified. Catalog stereotypes are automatically used behind the scenes.

Repositories

- Playpen now contains template folder structure to start new project and to ease solution architecture document generation.

Language

- Added connections in metamodel
 - Data Store – *Aggregation (is part of)* -> Application Component
 - *Organization – Access* -> Material
 - Material – *Access* -> Organization
 - Application Function, Application, Application Component and Data Store – *Dependendy* -> Technology Function

Miscellaneous

- Property Window always shows the actual stereotype name
- All generated diagrams are independent of the database technology (and can be opened for ex. after project transfer)

B0316

Productivity tools

- Improved diagram generation independent of database type.

- Generated diagrams include links to other diagrams under the same element.

Repositories

- Script for upgrading repositories from v2.x and v3.0 to v3.1.

B0315

Core capabilities

- Support for very large organizations and repositories
- Any number of organizations (e.g. subsidiaries) can now be stored in one single repository. Dashboards and reports can select any or all.
- Different (sub-)organizations including projects can hide content and share content

Repositories

- All catalogs (Applications, Roles, Organizations, Processes, System Software...) can be spread across any number of stereotyped packages with a distinguished icon.
- Location of language metamodel element is now a stereotyped catalog. No more GUID to be specified.

Dashboards

- Predefined dashboards (and charts) in the startup repository.

Language

- New diagram type called "Element Prototypes" providing access to all element and connector types.
Used for
 - Metamodeling
 - Defining the types of element for which tabular reports (Excel and CSV) or automatic value calculation are needed.
- Added tagged values for Epic: Added Start_Date and End_Date
- Added connections in metamodel
 - Node Instance – *Dependendy* -> Node Type
 - Application Function, Application, Application Component and Data Store – *Dependendy* -> Technology Function

Productivity tools

- The scope for calculation, tabular report generation, validation and diagram generation is automatically calculated. No more SQL statement needed.
- Import adds tagged values if they don't exist
- Empty element names are automatically replaced by '?' except for decisions (for consistent reporting, filtering & dashboarding)
- Validation engine correctly sends error messages to individuals following their assignment to functional domains

B0300

New Packaging – Features and Pricing for Different Audiences

See "Products" on www.labnaf.one

Parameters for Automatic Calculation and Excel Generation RULES

- Define your own parameters in the repository and use these parameters in SQL statements.

Language

- Added N-Ary Association. Used in Entities and Information viewpoints, constraint, resource

Labnaf AddIN

- Calculation: You can request immediate calculation for a single element, for a single calculation rule set or for all elements of a certain type (right-click on any of these elements and select the calculation option). Tagged value is added if it does not exist.

This is also very convenient for adding calculated tagged values to be used in charts.

- Diagram generation, calculation and tabular report (Excel) generation can be started by selecting an element in the project browser or in any diagram.
- Import Excel and CSV on demand with or w/o field mapping tables
- Generate Excel and CSV on demand following tabular report template (model)
- Labnaf cache: Work off-line with Labnaf using last version of language and rules even when creating new empty repositories.

C-Level ENterprise Visualization & Navigation

- C-level dashboards, charts, enterprise visualization and navigation with [Prolaborate](#).

Customization Workbench

- Connector Quick Linkers: [Simple configuration of verbs](#)
- Various improvements

Language Transformer

- Transformation templates and additional examples

Documentation

- Posters integrated in on-line guidance: “Modeling elements and connectors”, “Viewpoint classification”.

B0204

Repositories

- Sample repository
 - End to end navigable example (BOPCO: The Bus Operating Company) including
 - [dashboards](#) and especially [solution architecture completion dashboards](#).
 - generated documents
 - ...
- Mandatory [viewpoints scenarios](#)
 - A project needs comply with a specific solution architecture scenario.
 - The projects dashboard shows the completion status for each project
 - If you click on one of the projects, you can see, for that project
 - what viewpoints are expected
 - which viewpoints are missing
 - how many are missing (with auto-coloring depending on the number)
- All repositories: Word document assembly for Principles document and Standards document generation

Language Transformer

- Simplified and extended language transformation capabilities
- Documented language transformation development lifecycle
- Full set of language transformation examples

Metamodel

- Predefined environment for your custom metamodel
- Default Labnaf metamodel
 - [Language metamodels documentation](#), including
 - mapping of the metamodel to the transformation process
 - sample metamodel evolution
 - your metamodel step by step
 - related viewpoints needed
 - Additional connections for solution architecture
- Ability to use the Labnaf dynamic metamodel features with ArchiMate

Configuration

- Core configuration element renamed to “(((ModelingLanguageConfiguration)))”.
The version number was removed as it was unnecessary.

B0203

AddIn

- Process: Added “Automation” tagged value

B0202

AddIn

- Generate Tabular Report on demand (right-click on any Tabular Report template)
- Floating licenses managed by Sparx Systems Keystore Service

Repositories

- Sample Repository
 - Added different kinds of sample charts
 - Added [template content for project scope](#)
- Repositories documentation
 - Added links to EA documentation and sample files to illustrate how to import new elements from CSVs

Changes to the Default language metamodel

Added

- Application, Application Component, Data Store, Equipment and Material --is part of --> Representation
- Entity, Role and Process --is owned by--> Functional Domain, Area, Block, Category and Service

PowerShell

- ImportCsv creates tagged values even if they have not been defined in the modeling language

B0201

Labnaf Guidance Model Repository

This is a new item added to the Labnaf product list. It is the source model repository from which the entire online guidance is generated. It is provided for free with a license of the Labnaf Server.

EA's user experience Optimizations for Labnaf

On startup, Labnaf Addin now always optimizes EA's user experience for Labnaf as follows:

- Show the Properties Info Section

- Hide Stereotypes in the Project Browser
- Disable fully scoped object names
- Disable Auto Instance
- Disable Automatic Sub-Activities
- Auto Group Elements OFF
- Prompt for connector delete
- No Strict Connector Syntax

Changes to the Default language metamodel

Added

- Logical Node --Realizes--> Technology Function
- Logical Node --Network Interlink-- Communication Network (*needed when Logical Node is a firewall*)

Tagged Value Changes

Default value for tag "Form Factor" = "Virtual":

- Logical Node and Communication Network are now virtual by default.

B0200

Added [Labnaf PowerShell](#) featuring:

- Job scheduler
- Diagram generation following Labnaf diagram templates defined in the Labnaf model repository.
- Excel generation following Labnaf Excel templates
- Word generation following Sparx System's document template approach.
- Publication to an HTML site where pages can be shared using standard e-mails.
- Automated tagged value calculation following configuration defined in the Labnaf model repository.
- Invalid connector detection and email error reports distribution
- Backup to local model repository.

B0195

Added [Language Transformer](#) featuring automatic transformation, in a repository, of existing modeling language elements and connectors in terms of types, stereotypes and tagged values. Enables translation of any language e.g. ArchiMate, BPMN or UML, into any other language e.g. Labnaf (and vice versa).

B0191

Added Startup repository and Sample repository.

The startup model repository structure includes

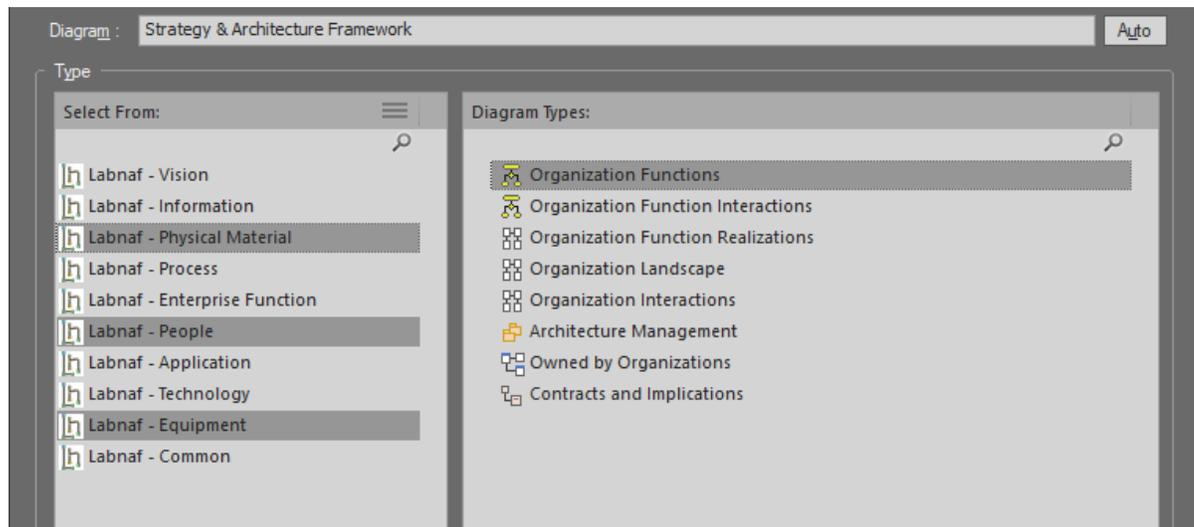
- A configurable language metamodel with dynamic alignment of the model validation rules
- Canonical and configurable repository structure of packages, elements and diagrams that one can copy from
- Configurable templates for diagram generation
- Model templates for configuring value calculations
- Reusable diagrams and elements (auto-legends...)

The sample model repository structure includes samples models and a canonical folder structure. Content of the canonical folder structure, at any level of detail, can be cut and paste into the production repository.

B0190

The [strategy and architecture content](#) organization has been semantically purified

- The language is richer in terms of vocabulary. You can now do precise modeling of IoT and ecosystems.
- [Viewpoints](#) are very consistently organized following a common refinement approach
- Toolboxes have been grouped accordingly



- The [language metamodel](#) has been extended and thoroughly organized into different perspectives.
- Architects configure the metamodel in one click and generate the validation rules in one other click
- Everything is now precisely [aligned to the original systems semantics](#).

B0180

The language [customization workbench](#) has been much improved and is now also configured in one click.

