



arKItekt 4.4.3 Release Notes

arKItekt 4.4.3 implements new features:

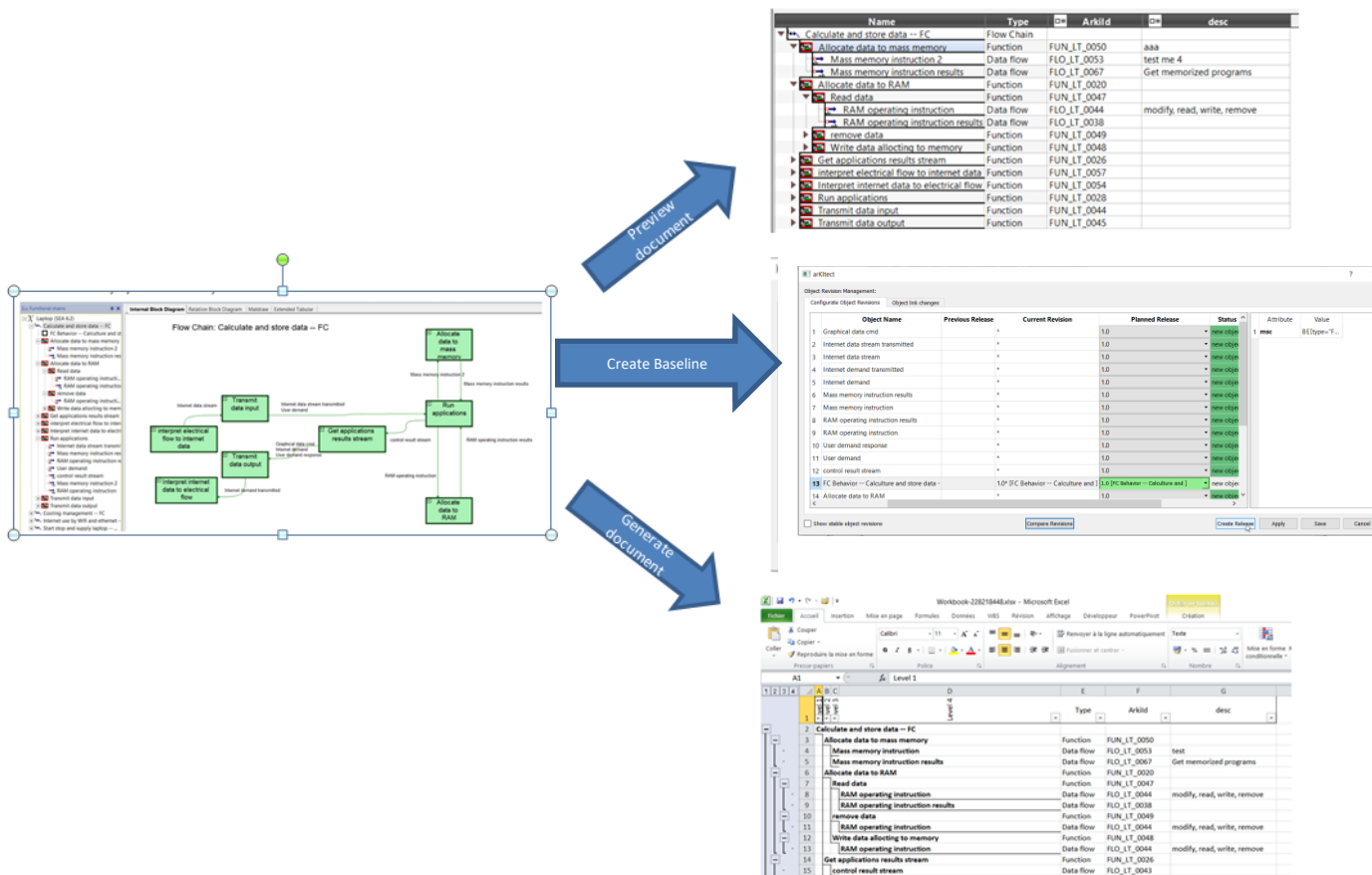
- Through Chains, we propose to **baseline a part of your data and generate Excel or Word document**,
- Create a network to group and propagate data between components thanks to the new **Message**,
- Save your graph view, restore it to **undo a graphical changes**,
- Control **impact analysis** and apply changes of the model **to the Chains**,
- **Merge** a project take account of **object revision**, resolving conflict is easier
- **Model gateway** get several improvements:
 - Show all objects imported to a **Chains**,
 - Export **objects variants**,
 - **Scan several time** the same sheet and **define order** of sheet/section
- insert **images** to **Markup** attribute,
- List all **object history** : control object modification (change of attributes, allocation of object ..) thanks to a python API
- Designer: copy a **rule configuration** to another branch

Baselining	2
Message mechanism	3
Graphical undo	4
Impact analysis of model changes in a Chain.....	6
Diff and merge: Merge project.....	7
Model Gateway improvements.....	8
Insert images to Markup attribute	9
Object history API.....	9
Designer: Copy a rule configuration.....	9



➔ Baselineing

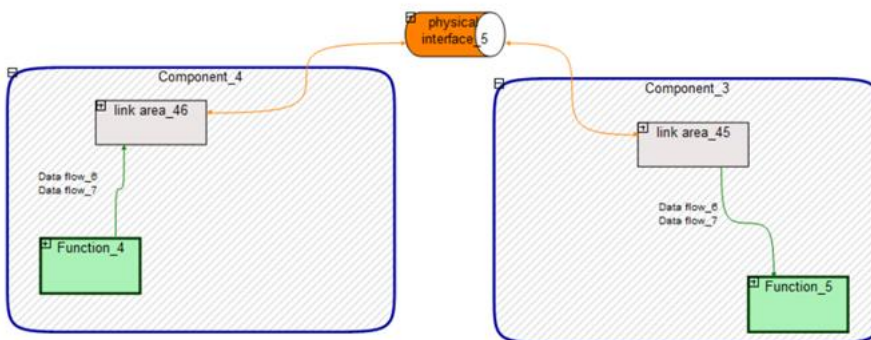
- ➔ There is now a dedicated mechanism in arKItekt allowing creating baseline through Chain content. Use Chain mechanism to construct the document elements, show only objects and flows what you want to manage in the configuration, customize your document format (hide or show object attributes, show object as an attribute ...), control objects revision to be in the document, integrate model modification to the Chain



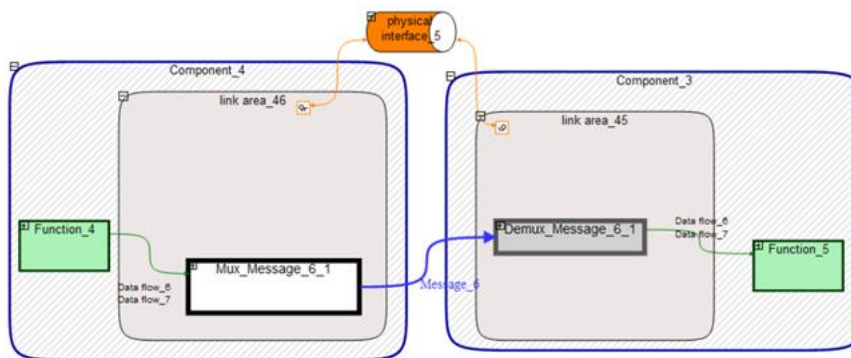


→ Message mechanism

- Build a network easily thanks to the new Message mechanism. Propagate data through the network, construct automatically the mux and demux that host messages and data sharing between two components.



Propagate data to link area by copy them into interface

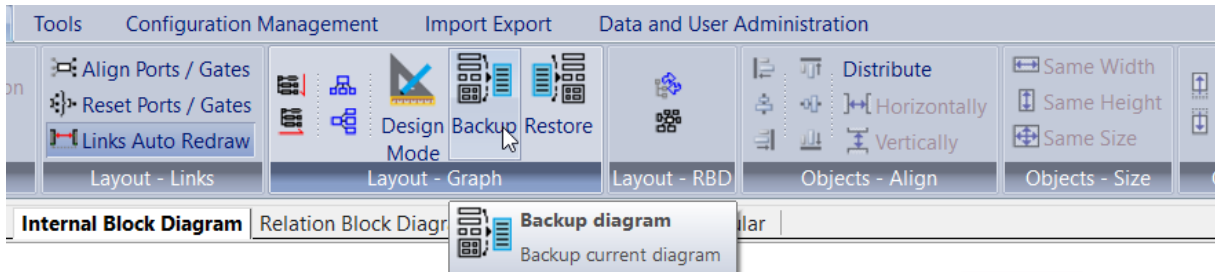


- Automatic creation of mux/demux on adding message flow
- Propagation of data in the network on allocating data to message

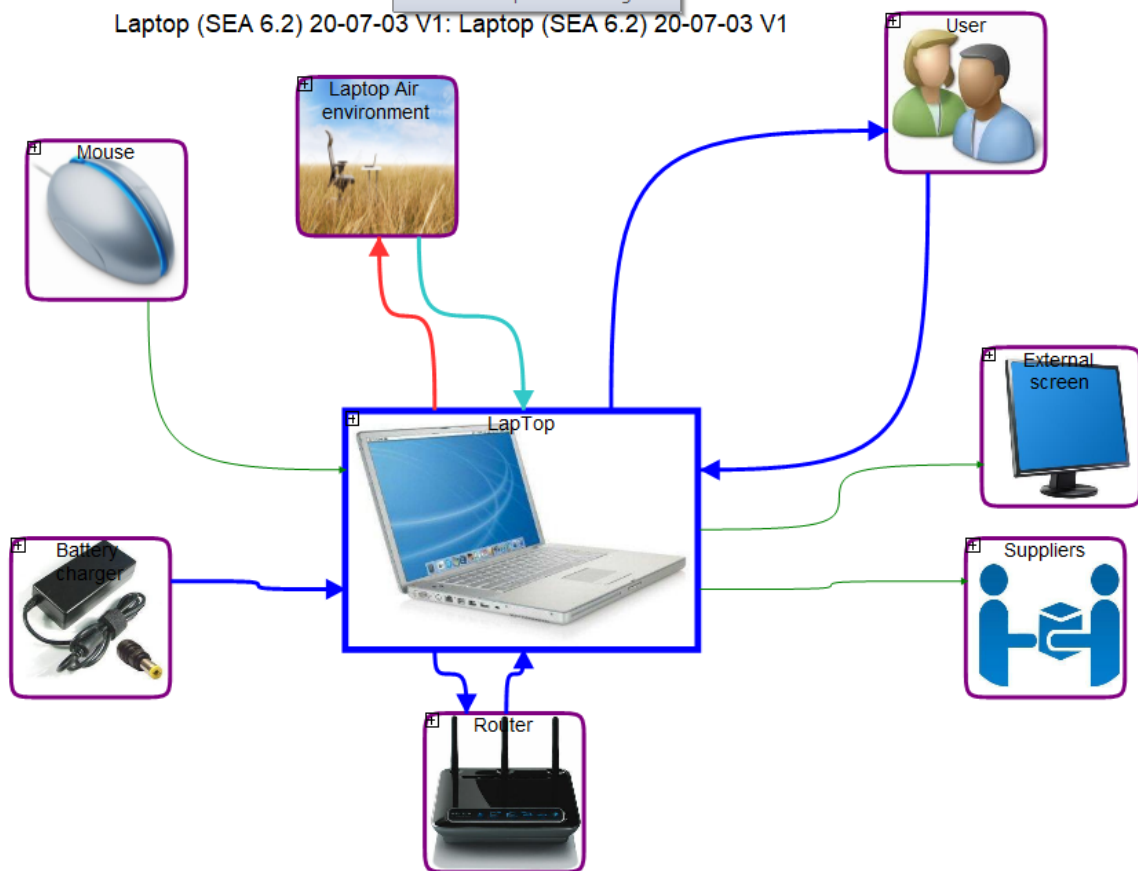


→ Graphical undo

- Backup all objects position and restore it when you want



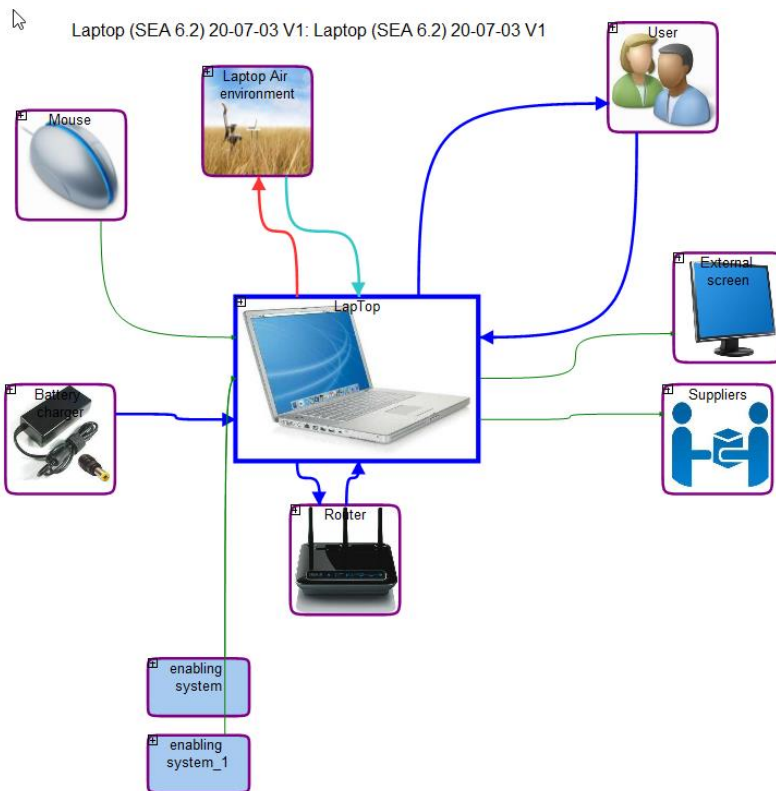
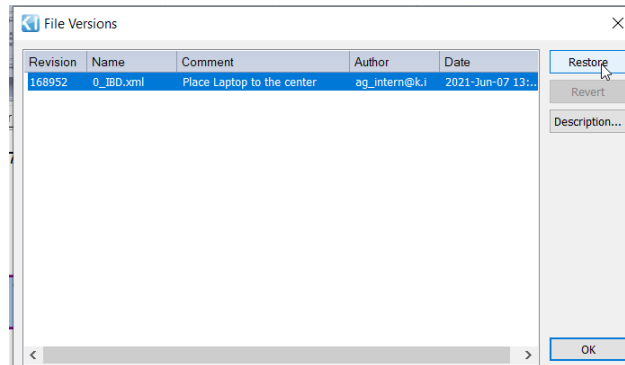
Laptop (SEA 6.2) 20-07-03 V1: Laptop (SEA 6.2) 20-07-03 V1





Knowledge Inside
arKItekt your system

→ Revert graphical move of objects if your view has been broken.





→ Impact analysis of model changes in a Chain

- When a model is updated, it can have some impacts to a Chain: Chain need to be updated too to include new objects and new flows. We introduce a new script “check chain”

The screenshot illustrates the workflow for impact analysis. On the left, a menu shows 'Merge chains', 'baseline', and 'check chain' (highlighted). A blue arrow labeled 'Display all changes impacted the Chain' points to the right. On the right, a dialog box titled 'Please validate chain objects:' contains a table with the following data:

	MetaModel Type	Object Name	New chain candidate	Ignore	Add to chain
1	Function	Transmit data output	New Function child(Function)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	Function	Transmit data output	New flow(Data flow)	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Below the table, a blue arrow labeled 'Add or ignore object and get result' points to a tree view on the left. The tree view shows a hierarchy of objects under 'Calculate and store data -- FC', including 'FC Behavior -- Calculture and store data -', 'Allocate data to mass memory', 'Allocate data to RAM', 'Get applications results stream', 'interpret electrical flow to internet data', 'Interpret internet data to electrical flow', 'Run applications', 'Transmit data input', 'Internet data stream', 'Internet data stream transmitted', 'User demand', 'Transmit data output', 'Graphical data cmd', 'Internet demand', 'User demand response', 'Internet demand transmitted', and 'New Function child'.

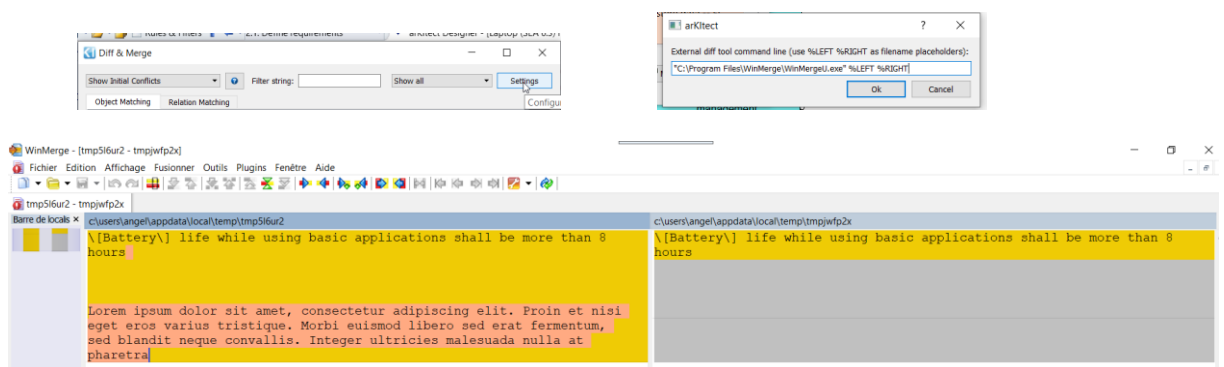


→ Diff and merge: Merge project

→ Merging a project with a dump take account of :

- object revisions
- Variants
- Phases

→ Configure diff & merge to use a diff tools





→ Model Gateway improvements

- Import data to a Chain: Import or update easily a chain thank to Model Gateway import. Model Gateway now allows importing and directly showing objects in a Chain. Thus all objects that is listed in the reference excel file will be visible in the Chain.

Sheet: Chain_2

Make some sheets to define object relation in the Chain

	A	B	C
1	Chain	parent function	child function
2	Users interaction -- FC	Display image	New function child
3	Users interaction -- FC	New parent function	other function child
4	Users interaction -- FC	New parent function	New function child
5			

Sheet: Chain data

Need a specific sheet with all object be added and shown

	A	B	C
1	chain	object name	object type
2	Users interaction -- FC	New function child	Function
3	Users interaction -- FC	New parent function	Function
4			
5			

- Export object variants: Up to now, arKItekt and Model Gateway support only to export object options. It is now possible to export object variants: each variant where object is visible will be listed in the generated file.
- Define a sheet order to import data: When we try to import complex excel file which use some object definition and relations in different sheets, it may occurs some conflict while importing them to arKItekt. Solution is to define an order between sheets to be imported.
- Scan several times the same sheet: Create a rule maker model can be difficult and it may be easier to 'split' a sheet in different section. Besides, when we try to import a complex excel file which use some object definition and relations in different sheets, it may occurs some conflict. Solution is to define an order between sheets/section to import data.



→ Insert images to Markup attribute

- Markup attribute has been introduced with arKItekt 4.4.2. It is now possible to insert image in such attributes

→ Object history API

- There is a new python API **GetHistory** that allow to list all actions made on an object. Thus, we can control object lifecycle: who change attributes values and when.

→ Designer: Copy a rule configuration

- Copy a rule configuration (all rule attributes and all children rules checked status) to another branch)

