



arKItect 4.4 Release Notes

arKItect 4.4 implements major new features:

- It is now possible to filter views according to **product lifecycle**: it is possible to see a system in the context of a particular phase allowing verifying consistency of your model for that phase,
- **Variants and options** is improved and we eventually got a **propagation strategy** allowing simplifying drastically the job off tagging object options,
- A new kind of attributes is now available in arKItect: **wiki attributes** make arKItect close to a formal wiki with all its advantages in terms of flexibility, references to objects...,
- We propose now an **implementation of parameters** with ability to reference parameter values in requirements descriptions for instance, but it is also possible to link any object to a set of parameters and their values, now with a single requirement, you can cover many variants,
- Through functional chains, we propose a new automated strategy for **impact analysis** of the changes in your system or your project, a strong support to the key activity of any system engineer or project team,
- arKItect now supports a mechanism of ownership submitting object modification to a workflow, preventing some objects to be modified without notice.

| | |
|--------------------------------------------------------------------|---|
| Product Lifecycle | 2 |
| product line management of arKItect models | 2 |
| Wiki attributes..... | 3 |
| Parameters and object references..... | 3 |
| Generalized Impact Analysis | 4 |
| Ownership | 5 |
| Generation of a tabular view for a multi-selection of objects..... | 5 |
| Location for a multi-selection of objects..... | 6 |
| Other improvements..... | 6 |
| Fixed issues..... | 7 |

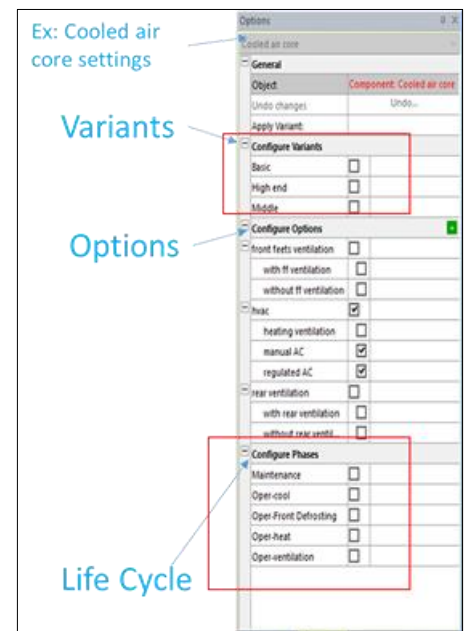


➤ **Product Lifecycle is now a first class citizen in arKIitect**

- Product or Service lifecycle is a key aspect of Systems Engineering. There is now a dedicated mechanism in arKIitect allowing to define the phases in which an object should exist and filter system display per phase.

We also introduced a propagation mechanism adapted to phases and based on functional chains. In fact a phase is a functional concept in which all system parts interact consistently. It was then logical to rely on functional chains to define a system per phase and propagate phase information efficiently.

- Lifecycle is defined as an independent dimension from diversity (options and variants). It is then possible to visualize your system for a given set of options and a given phase.



New lifecycle and options window

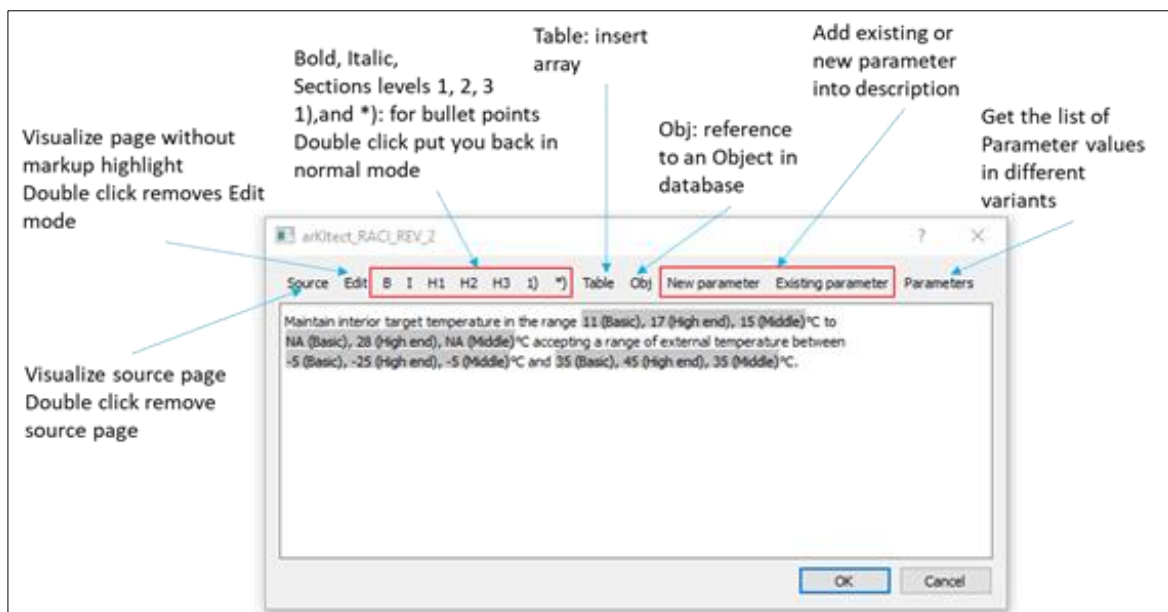
➤ **Breakthrough in product line management of arKIitect models**

- Up to now, options in arKIitect suffered a limitation (although this fits with 95% of the cases actually met). It was not possible to attached different combination of options to a single object. This is now fixed and the diversity definition in arKIitect has now the logical power of a full configuration management system.
- A strong and powerful innovation has been brought in parallel for options and variants propagation. We got a deep understanding of the relationship between options, variants on the one hand and hierarchy and allocation in systems engineering on the other hand. This led us to introduce a segmentation between objects for which options are manually defined and objects which may inherit options from their “position” in a systems definition. We could also identify sound options propagation rules, something we were thinking about since 2011, when options and variants were first introduced in an arKIitect release.
- It is now possible and sound to propagate about 95% of the options information into a project. This is a major productivity breakthrough for managing product lines. It is also possible



➔ **Wiki attributes: better than text, cleaner than rtf**

- ➔ Up to now, arKItekt did support memo and rtf text attributes. The memo format is a basic text format forbidding styles, character size, insertion of objects like images. On the other hand rtf format allows doing everything but is rather complex to handle. arKItekt now supports a wiki format for textual attributes.
- ➔ It is much more powerful than previous formats
- ➔ A key innovation is that you can refer to an object in a wiki attribute and this is a new kind of link between objects in attributes (after hierarchy and messages). Reference to objects has been introduced to manage another great innovation: parameters.



New Wiki attributes now support reference to parameters and their values depending on variants.

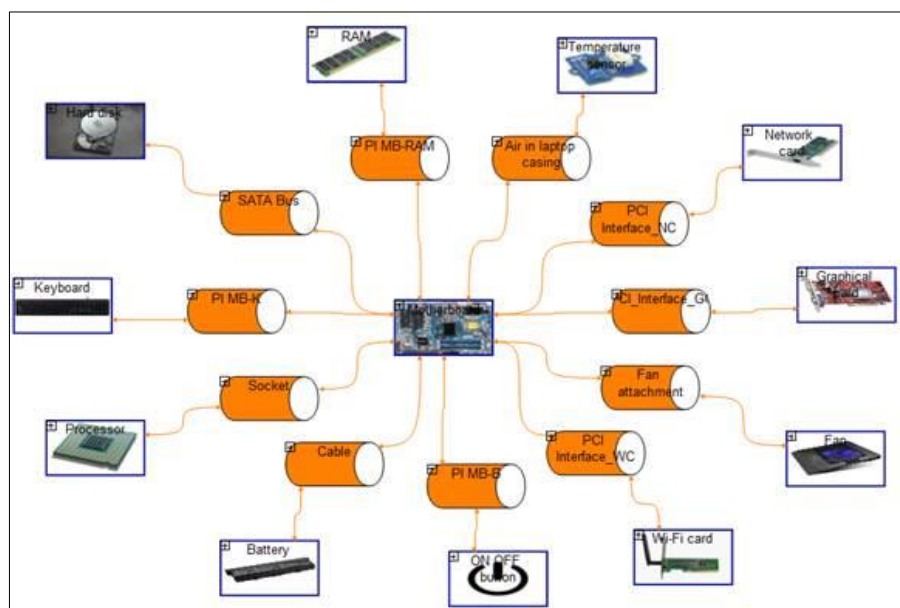
➔ **Parameters and object references in wiki descriptions**

- ➔ Parameters are the missing link between Systems_Engineering (including requirements management and system architecture) and System Engineering (simulation of control and physical systems). They also correspond to “criteria” in Functional Analysis. Most of the time they are simply not identified during requirements analysis but each time some performance or number appears in a requirement, it corresponds either to a constant or a parameter. Such constant and parameters define the system performance.
- ➔ arKItekt now supports a native parameters and parameter values dictionary. Any object may refer to a set of parameters and parameter values in their wiki attributes.



- ➔ Parameter values and variants configuration management: the same requirement shall have different revision depending on variants where its related parameters values are different. This complexity is now supported in arKItect revision management mechanisms. This allows sharing the same requirement object with different parameter values depending on the variant we consider. For instance, one may consider a max speed requirement with value 220 km/h for a high end vehicle and 165 km/h for a more basic configuration.
- ➔ **Generalized Impact Analysis**
 - ➔ Impact analysis is a primary concern of our customers. In arKItect, you can manage consistency and completeness of your requirements and architecture. This is much stronger than traceability alone. Once you use arKItect, you get a consistent and complete system model which is a good basis for generating automatically documents. However, the next hot issue is change management and here comes impact analysis: certainly the most important feature for systems engineering and project management.
 - ➔ You can now build an impact analysis on any object or set of objects in an arKItect database: arKItect produces a functional chain around your object of interest (whether a requirement, a function, a flow, an interfaces... or a set of such objects) and investigate all related objects through all possible type of links or flows at any depth and in any view.
 - ➔ If you agree that the eventual purpose of modeling in Systems Engineering or Project Management is impact analysis, this is simply the Holy Grail and it is now available as a basic functionality for any data model in arKItect 4.4.

Impact analysis for a change on a component in a systems engineering model, but you can do this for functions and flows, requirements, processes...





➤ Ownership

- While the spirit of arKItect is opened collaboration in a team, some customers asked us for a long time means to prevent some object from being modified without sufficient authorization level. Up to now, various approaches allowed a curative answer to this requirement allowing to identify and revert unexpected changes (Undo, objects history, objects and project versions). We now propose a mechanism preventing objects from a selected scope to be modified without agreement from a person accountable for that scope.
- a modification workflow is now available in arKItect 4.4 for controlling model changes.

| Date | Status | Actor | Object Name | Object Type | Action | Details | RACI details | Decision |
|------------------------|-----------|--------------|-------------|-------------|---------------------------------------------|---------|-----------------------------------------------------------------------------------------|--------------------|
| 2019-09-17 09:38:38 | Submitted | ag_marta@k.i | R212 | Requirement | Set attribute [desc] for [Requirement]R212' | | Waiting for decision: Accountable_User2@k.i Accepted by: Accountable_User1@k.i | Accept/ Decline |

Filters: [] [] []

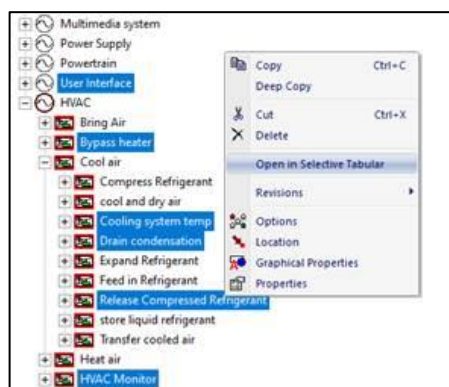
Showing 1 to 1 of 1 entries

Previous 1 Next

Example of request

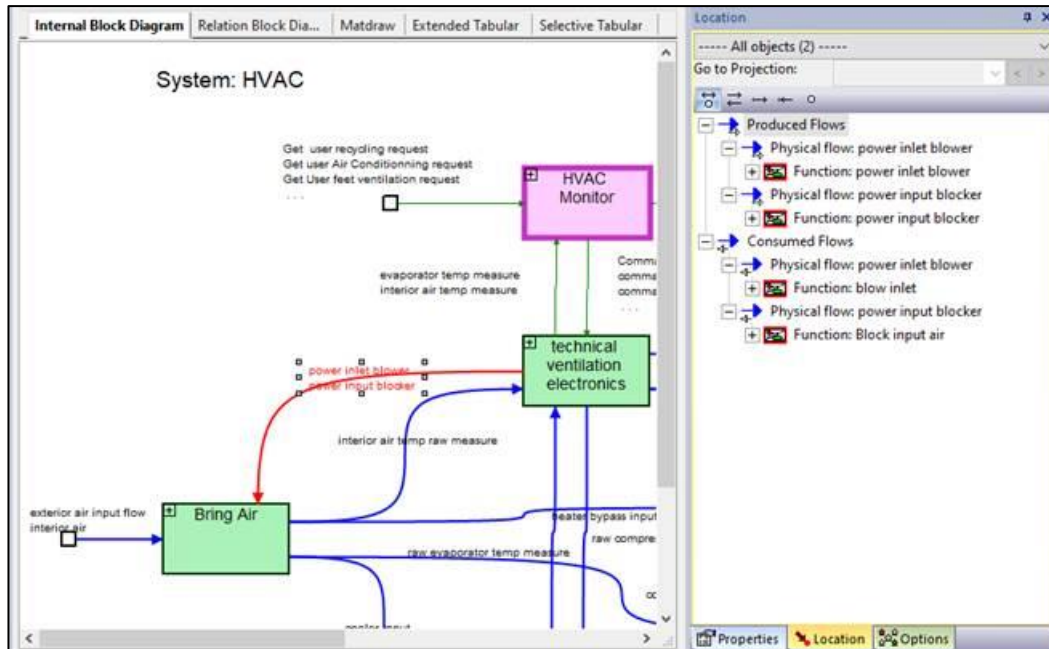
Generation of a tabular view for a multi-selection of objects

In any arKItect view, you can now select a list of objects of different types and build a tabular view with these objects allowing selective import/export in Excel or attributes changes in tabular view.



Location for a multi-selection of objects

It is now possible to get the location for a set of objects. This is particularly useful for flows containing several data. You can now visualize at one the location of a set of objects and drag&drop what you need in functional chains. In the past, it was only possible to do this for one object at a time.



Other improvements

- Location: a script is available to display only chains in which an object actually appears
- Number of selected objects in a multi-selection is displayed
- Drag&drop an object directly from a Search window.
- Merge chains can be also launched from treeview

Fixed issues

- Resizing the Treeviews list window in menu "home" could lead to arKItect not responding. This is fixed
- Setting combination of options for objects is now possible. New combinations were not saved previously.
- In case of using a square selection for a set of objects in an IBD view and resize, there were graphical problems with the expand buttons ("+").
- Search settings are kept even when user stops an on-processing search.
- Increased memory and address space for arKItect process and python processes. Max size has been increased from 1 to 3GB.
- Improved collision management between sessions if user tries to copy/move an object under an object that has been deleted by another user
- Read-only hyperlinks are now clickable
- 255 characters limit for windows file path is now managed



Knowledge Inside
arKItect your system

- Overwriting a revision number is no more possible: in arKItect previous releases, coming back to a previous version of an object was possible and then it would be possible to modify it and make a new release. The release numbering was not prevented from reusing already used numbers. This is now fixed. Example if an object is in version 1.7 and you want to come back to version 1.3 and make changes and then make a new revision, this new revision should be 1.8 by default.
- Deep copy was not taking into account flows that were both produced and consumed by an object, this is now fixed.