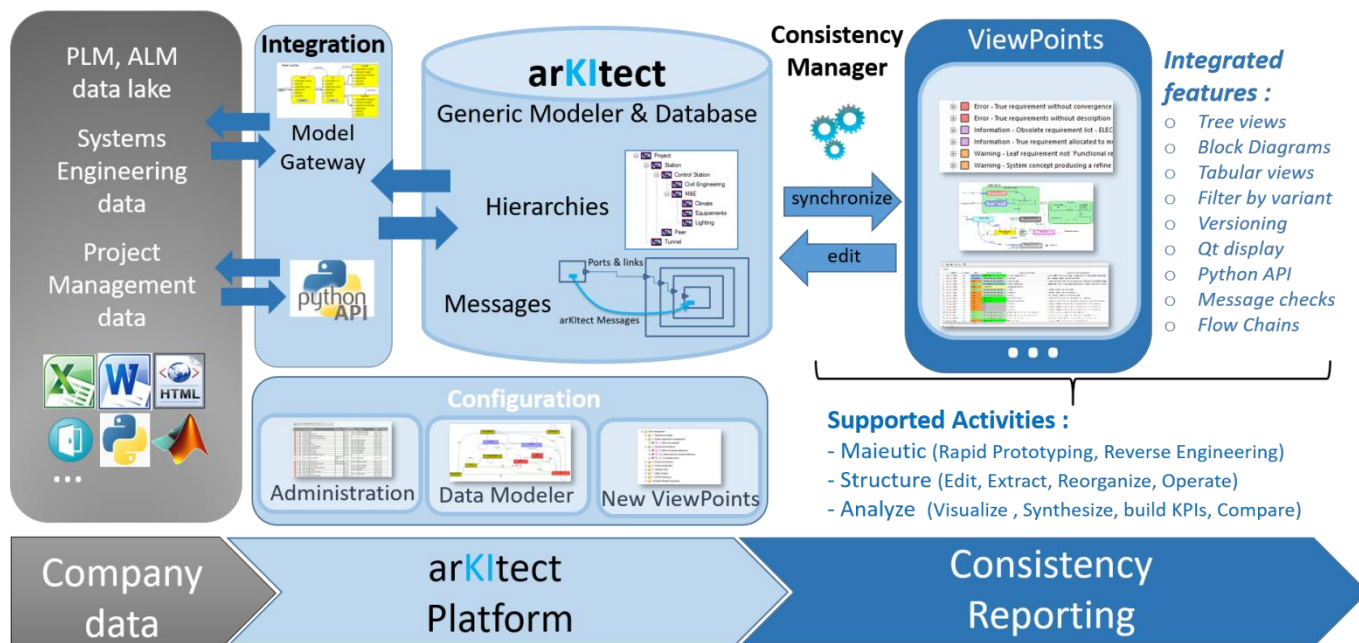


arKitect

Designer & Developer

arKitect is a data integration platform enabling structuring, visualizing, and operating complex set of data stemming from various legacy sources (Excel™, database, web sites, xml...) or edited directly through arKitect collaborative interface.



Key Features

- Fast Data Model and Viewpoints development
- Generative and Synchronized Diagrams
- Low code configuration
- Generic Functional Chains
- Collaborative Undo/Redo
- User Administration
- Options and Variants management
- Extensive python API
- Objects and Project revision management
- Opened : import/export all your data



This information booklet applies to arKitect 5.1.0 release
arKitect is a trademark of Knowledge Inside

Knowledge Inside
arKitect your system



Data Model and applications design

Flexible Data Model Definition

arKitect allows you to design very easily your own *Data Model*: objects, flows and their relationships. From a Data Model you can develop an *application* by adding Python scripts operating on the related model.

arKitect modeler & database technology provides advanced support for handling hierarchies and interactions. Because systems are made of (sub-)systems in interaction, our technology perfectly fits for complex systems modeling or organizing complex sets of data whatever the application field.

arKitect includes an advanced visualization system of your data: you can easily produce generative views of the content of your database: graphical views of interactions (diagrams), tree-like representation, tabular representations.

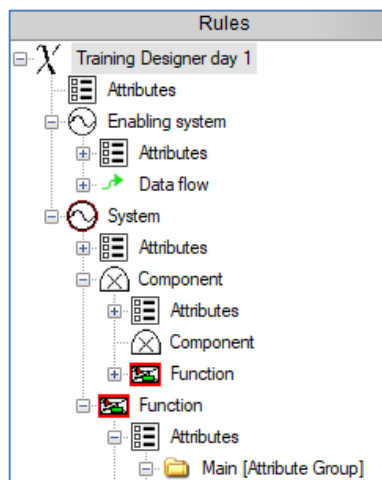
Rules

The basic entity in **arKitect** is an *object*. An object is defined by its Name and its Type. Types are important because they define the different kinds of objects that can exist in the database, appearance of the objects and relations between objects.

Types constitute the Data Model. **arKitect Designer** provides a window to edit a Data Model which includes following features:

- ✓ Add new types and delete existing ones
- ✓ Add hierarchical relations between Types – aggregation in Object Oriented notation
- ✓ Modify the properties of existing types
- ✓ Change the relations between types, using the context menu or drag&drop
- ✓ Define relations as flows
- ✓ Control the visibility of flows
- ✓ Manage type attributes
- ✓ Define keys for types (e.g. IDs)

The Rules editor is used to define Types and structure them using hierarchy and messages.



Data Model upgrade

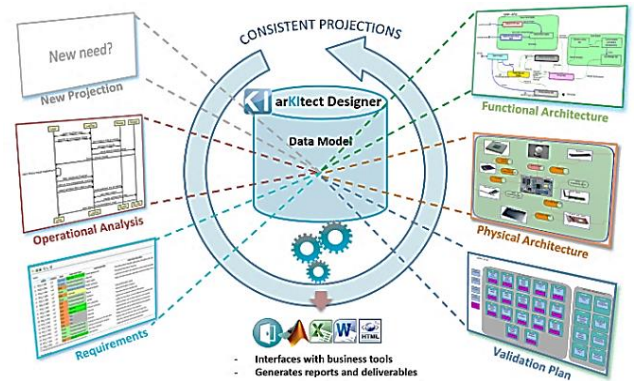
Once you define your first version of your own Data Model and deploy the related application, you will certainly identify new features, processes, hints that you want to implement.

arKitect MMU (MetaModel Updater) allows deploying Data Model upgrades on all your applications.

Generic Chains

Generic Chains are a powerful mechanism to describe pieces of architecture related to a particular topic. Chains provide additional filters representing partial views of the underlying database. When defining a chain:

- ✓ Objects in chains come with all their flows
- ✓ Show or hide objects, their children and the flows
- ✓ Merge several chains
- ✓ Drag&drop objects from « location window » enables performing an impact analysis based on propagation through flows



Generative and synchronized views of arKitect database

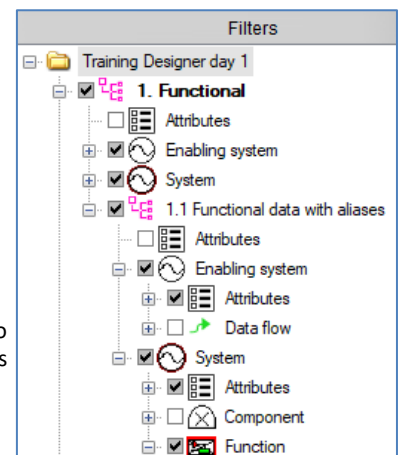
Filters

arKitect views are generative because they are a real-time filter on the content of the database. If some new objects are added to a database, e.g. through an import of new data, some related views will be modified automatically to display these new objects.

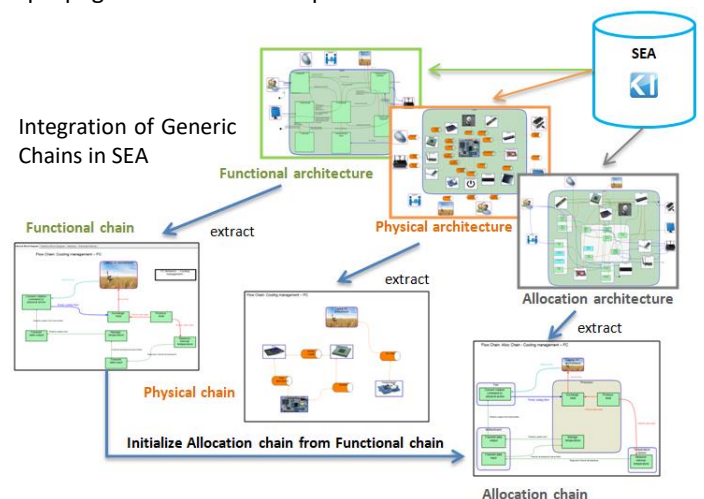
Filters are made of objects matching a subset of types of the Data Model. **arKitect Designer** allows you to:

- ✓ Define new filters and remove existing ones
- ✓ Copy filters
- ✓ Change the properties of existing filters
- ✓ Set a default filter
- ✓ Hide filters
- ✓ Apply powerful filtering principles
- ✓ Enumerate attribute in filters
- ✓ Create sub-filters

The Filters editor is used to create customized data views per domains and processes.



Filters can be used to edit manually diagrams of objects interactions and add/modify/remove objects, in which case their display is stored and the changes are committed to the database and propagated to all other impacted views.





arKitect interfaces

Message Sequence Chart (MSC)

arKitect provides a behavior description according to UML Message Sequence Charts (MSC). Actors and messages Types can be parametrized for any Data Model.

- ✓ Auto generation of MSC for any chain diagram
- ✓ Synchronization between IBD and MSC diagram
- ✓ Possibility to insert any object (e.g. Requirement) under any actor

Model Gateway

The Model Gateway (MGW) is a tool to build custom import/export of tabular data to/from **arKitect** models through a generic model transformation mechanism.

The model transformation is itself graphically specified in an **arKitect** model from which we can:

- ✓ Browse an **arKitect** database with MGW requests made of elementary operations (parent of, child of, input of,...)
- ✓ Import Legacy datasets to an **arKitect** database (content, structure, options,...)
- ✓ Extract a piece of Information from **arKitect** based on a MGW request
- ✓ Update legacy data imports for the sake of synchronization
- ✓ Perform a faithful import of legacy Excel™ file structure
- ✓ Configure export templates that can be packaged for end-user
- ✓ Add conditions on MGW requests (keys, constraints, ...)

MGW Advanced features

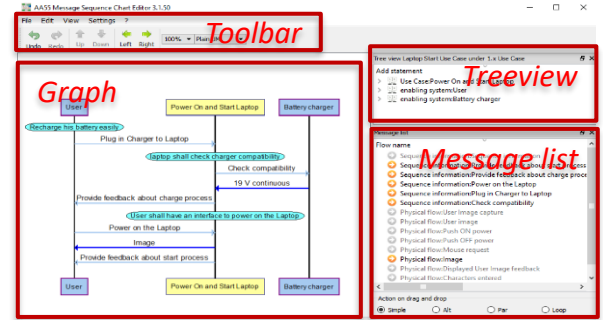
MGW request language and complementary modules enable:

- ✓ Indirect relations (ancestor of,...)
- ✓ Building integrated tabular views in **arKitect**
- ✓ Matrix formats
- ✓ Restricting MGW requests to a set of filters
- ✓ Restricting MGW request through syntactic rules (regex)
- ✓ Creating Excel™ templates with display and post-processing setup
- ✓ Applying MGW to the comparison of 2 versions of database (or 2 databases)
- ✓ Translating key in source data into an **arKitect** database key
- ✓ Hiding some relations at export using virtual columns
- ✓ Packaging MGW Model transformation into an **arKitect** application

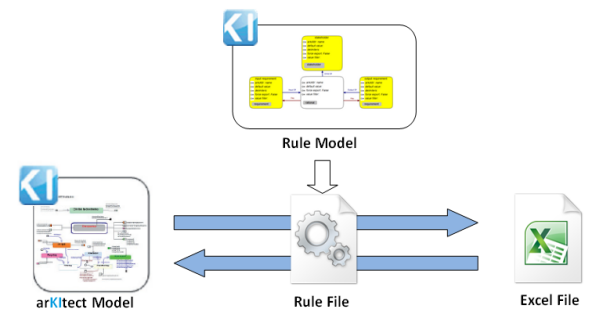
Python API

Act and interact on **arKitect** objects using utmost popular Python language. Any action available through the user interface is available through scripts.

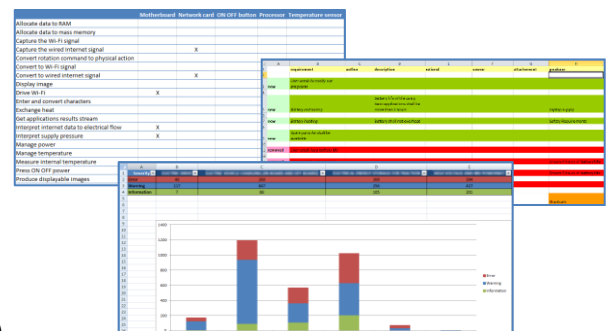
- ✓ Implement scripts directly delivered inside a Data Model
- ✓ Use dedicated libraries to browse and transform **arKitect** data
- ✓ Exploit the potential of Python community libraries
- ✓ Automate user action and provide a guided workflow environment
- ✓ Add custom GUI's to your application (PyQt)
- ✓ Interface with other tools using export/import capabilities



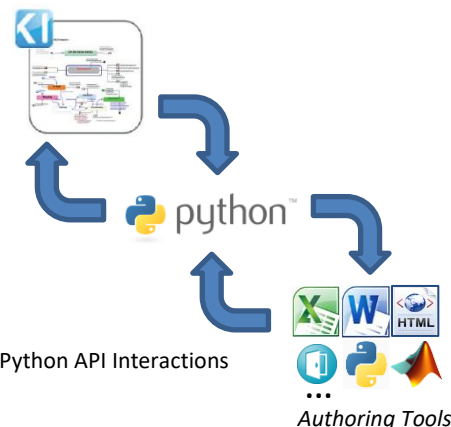
Specific MSC editor window



MGW – Interface principle



MGW – Example of features exported in Excel™



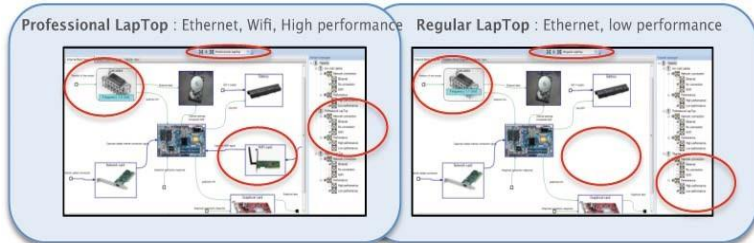
Python API Interactions

arKitect platform native features

Configuration Management

arKitect provides support for change and diversity management.

- ✓ Database version management
- ✓ Objects version management
- ✓ Diff and merge at database level
- ✓ Variants management transverse to the database:
 - Define options for objects
 - Configure variants for a set of options
 - Show/hide objects in views and viewpoints
 - Run all programs according to selected variant



Variants for product line management – show/hide objects depending on selected variant

Interfaces with other tools and formats

- ✓ IBM Rational DOORS
- ✓ Mathwork Matlab/Simulink
- ✓ ReqIF format import/export for requirements
- ✓ Embedded Model Gateway interface enabling
- ✓ XML and every format accessible through our python API

Licensing modes

- ✓ arKitect is a client-server platform
- ✓ It is licensed as a cloud SAAS application or on your own servers under yearly or perpetual license

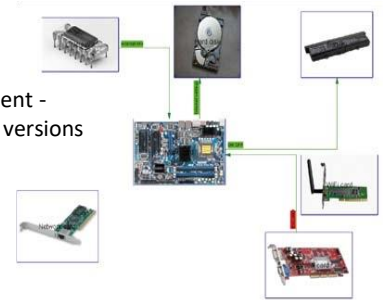
Commercialized arKitect Designer by-products

- ✓ arKitect SEA: manage your Systems Engineering data
<http://www.k-inside.com/web/sea/en/>
- ✓ arKitect WBS Builder: build your project management plan
<http://www.k-inside.com/web/wbs/en/>
- ✓ arKitect SoS (custom): build services or products

Key Advantages

- Intuitive tool with short learning curve
- Powerful import/Export technology
- Immediate hierarchical visualization of your data
- Guaranteed consistency of all data and their relationships
- Native traceability through data consistency management and revision management
- Agile datamodelling and scripting enabling rapid prototyping
- Programming interface, e.g. Automated generation of documents through extensive python API
- Automated update and consistency of all diagrams for all users after each user modification
- Return on experience and expertise of support team

Configuration Management -
Diagrams for comparing versions



Ergonomics

- ✓ Generative views: any object relation, properties and interfaces added to the database are automatically displayed in all views
- ✓ Graphical drag&drop for reorganizing architecture easily
- ✓ Expand/collapse objects to visualize children objects in diagrams
- ✓ Navigation: show location of an object, search by name, type, attributes... go to object location in any view and viewpoint

Collaboration

- ✓ Enable multi-access to the database and manage collisions
- ✓ Keep track of any modification by any user

User and data administration

- ✓ Define business space
- ✓ For each business space, define workspaces
- ✓ Each workspace may contain several databases
- ✓ Manage users' rights per workspace: read-only, modification, coordination
- ✓ Show/hide views at database level

arKitect Developer

- ✓ arKitect Developer allows deploying the data model designed with arKitect Designer to teams of end-users
- ✓ arKitect Developer is providing the same functions as arKitect Designer except it is not possible to access the data model definition windows (Rules and Filters definition)
- ✓ For an introduction to arKitect Developer browse our support site: <http://arki.k-inside.com/>

Available training module and services

- ✓ We propose a two days training module for arKitect Designer
- ✓ We provide advice about how to develop an application with arKitect Designer and a subsequent introduction to designing a data model and associated scripts as well as an introduction to Model Gateway and available python libraries.
- ✓ We also provide integration services and consulting either directly or through our partners



Knowledge Inside
arKitect your system

Tel. +33 (0)1 39 02 70 29
contact@k-inside.com
www.k-inside.com