

1. arKItect Developer Getting Started	2
1.1 Introduction to arKItect Developer	2
1.1.1 arKItect Developer Product Overview	2
1.1.2 Documentation Related to Developer	3
1.1.3 Starting and Quitting arKItect	3
1.2 First Steps	4
1.3 Developer Environment	5
1.4 Navigation in the Diagram	9
1.5 Navigation in Projections	15
1.6 Working with Objects	17
1.7 Management of Variants	28
1.8 Beyond the Developer Getting Started Guide	37

arKItect Developer Getting Started

Content

- Introduction to arKItect Developer
 - arKItect Developer Product Overview
 - Documentation Related to Developer
 - Starting and Quitting arKItect
- First Steps
- Developer Environment
- Navigation in the Diagram
- Navigation in Projections
- Working with Objects
- Management of Variants
- Beyond the Developer Getting Started Guide

Download



You can download this documentation and the tutorial in pdf format.

The aim of this Getting Started guide is to introduce you to the use of **arKItect Developer**. The presentation begins with an [introduction to the arKItect Developer](#) and to the related documentation. In addition, [Starting and Quitting arKItect](#) is presented.

The actual Developer tutorial consists of the following parts:

- [First Steps](#): opening a project and creating a copy
- [Developer Environment](#): presentation of the Developer user interface
- [Navigation in the Diagram](#): navigation in the hierarchical structure of the project
- [Navigation in Projections](#): navigation in different views
- [Working with Objects](#): creation of object instances, their modification and deletion
- [Management of Variants](#): creation of variants and options

In addition, [Beyond the Developer Getting Started Guide](#) offers a brief overview of the functionalities not presented in this Getting Started Guide (they can be found in the [arKItect Developer User's Guide](#)).

Introduction to arKItect Developer

This chapter introduces you briefly to **arKItect Developer** as well as to the related documentation. Starting and quitting **arKItect** are also briefly presented.

The chapter is structured as follows:

Contents

- arKItect Developer Product Overview
- Documentation Related to Developer
- Starting and Quitting arKItect

arKItect Developer Product Overview

arKItect Developer is a graphical tool for the conception and representation of complex hierarchical systems.

arKItect Developer Assets

The key strengths of **arKItect Developer** are:

- Hierarchical system representation
- Graphical system definition
- Multi-view visualization

- Variants management
- Added functionalities via Python scripts

arKitect Developer Environment

The user interface of **arKitect Developer Environment** offers you notably the following system conception and visualization tools:

- Internal Block Diagram and treeviews for the visualization of the hierarchical system structure
- Palette for the creation of new object instances
- Options and Variants Manager for diversity management
- Special views (Relation Block Diagram and Extended Tabular View) that provide additional information on the system
- Properties for the easy visualization of object properties
- Location for information on the object's place in the object hierarchy
- Options for the configuration of object options related to variants

Documentation Related to Developer

Accessing Documentation

To access **arKitect Developer** documentation via the **arKitect** user interface, go to the **Help** menu and select **Documentation**. You can also use the F1 keyboard shortcut.

Download Documentation

arKitect Developer Getting Started Guide can be downloaded in the form of a printable pdf from the [documentation homepage](#).

Developer User's Guide

Besides this Getting Started Guide, you have also access to the **arKitect Developer User's Guide**. The **Developer User's Guide** is structured as follows:

arKitect Developer User's Guide

Other Getting Started and User's Guides

You may also have access to the Designer documentation (depending on your license) and the Administrator's Guide:

- [arKitect Designer Getting Started and arKitect Designer User's Guide](#)
- [Administrator's Guide](#)

Other On-line Documentation

You may also wish to access the following complementary documentation:

- [Installation Guide](#)
- [Release Notes 4.4.0](#)
- [Scripting Documentation](#)
- [Glossary](#)
- [FAQ](#)

Starting and Quitting arKitect

When **arKitect** is installed, an *arKitect 2.1.x* folder should be created in the Windows Start menu. You can launch **arKitect** directly via the **arKitect**

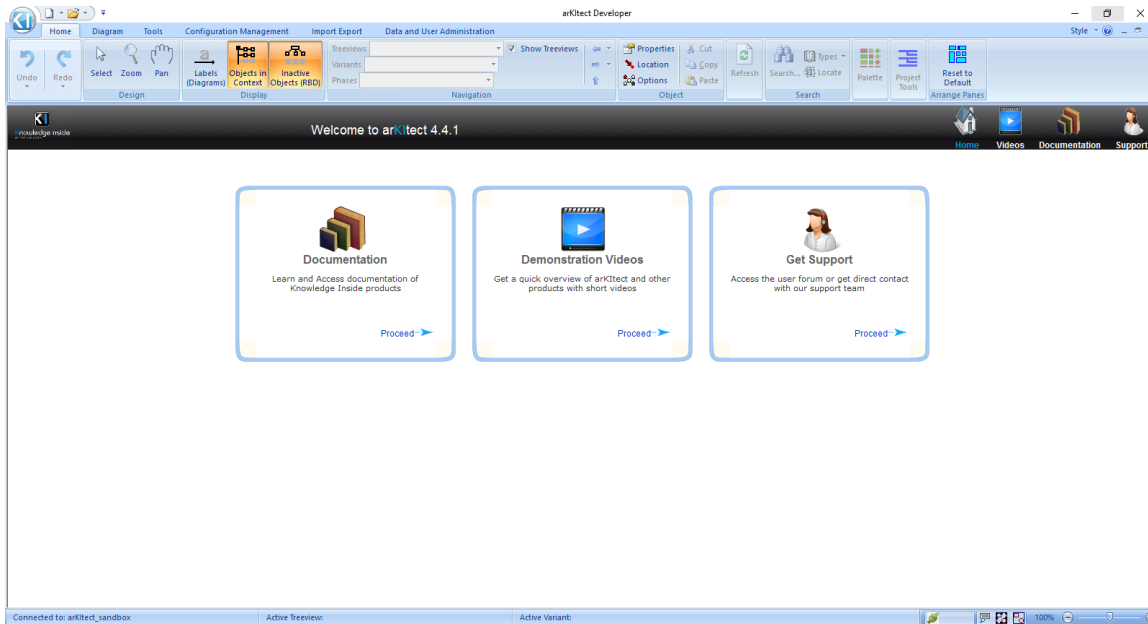
t icon  found in this folder.

To exit **arKitect**, go to **File Exit** or click on the cross at the top right-hand corner of the **arKitect** window.

First Steps

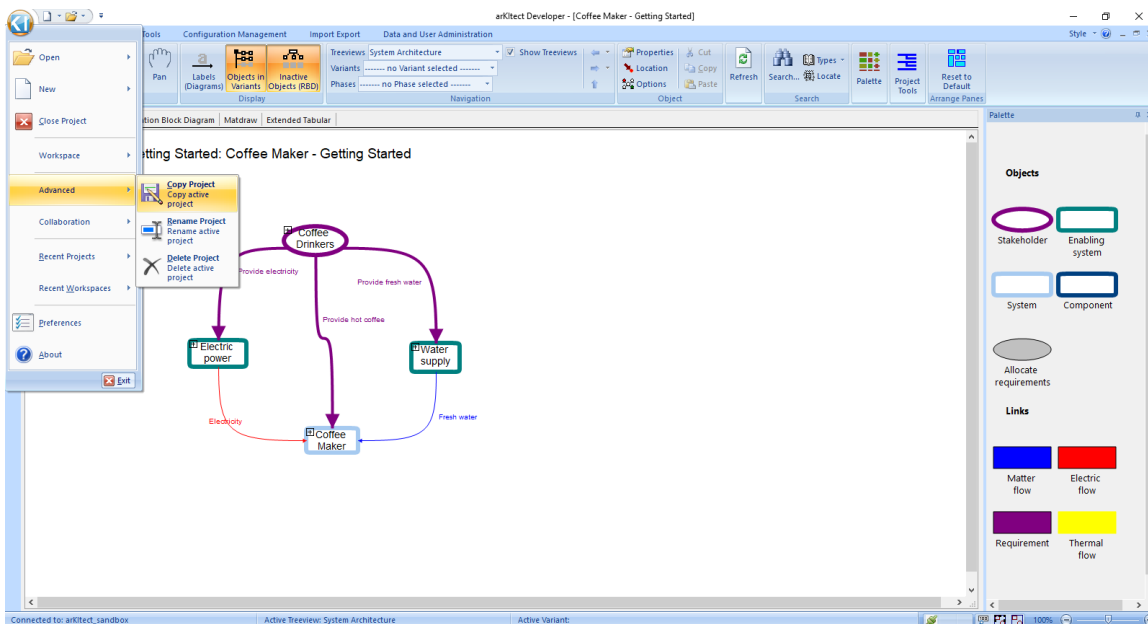
We begin this tutorial by starting **arKitect** and opening the **Coffee Maker - Getting Started** project in workspace **Demos**. This workspace is one of the workspaces created automatically for every user.

After starting **arKitect**, the Coffee Maker project can be opened by accessing the **Main button(File)** menu and choosing **Open** from the list of actions or simply using a **Ctrl+O** shortcut.

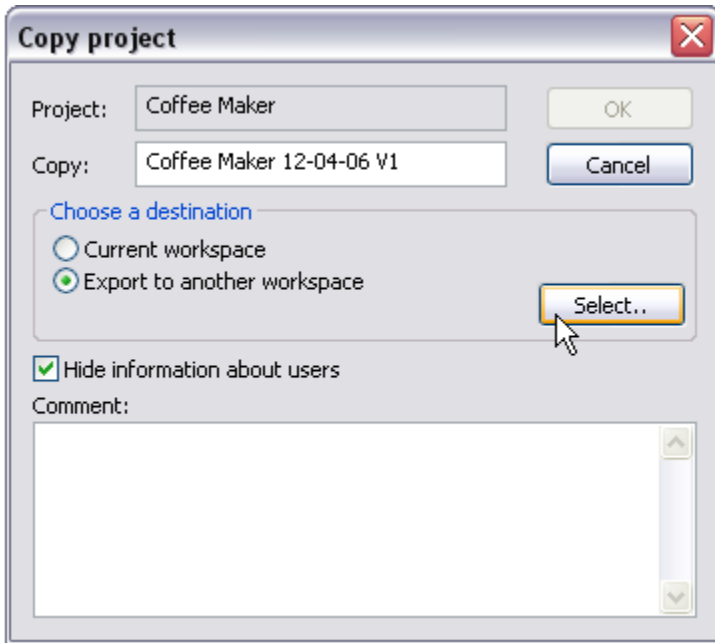


The Coffee Maker - Getting Started project opens on the following view. In this project, a generic coffee maker has been modeled. Both the physical components of the system (objects of type System and Component) as well as the Stakeholders and their Requirements have been represented.

In order to continue working on the project while conserving its initial version intact, we shall begin by creating a copy of it. Go to **Main button(File) Advanced** and choose **Copy Project** from the menu or simply use **Ctrl+S** shortcut.



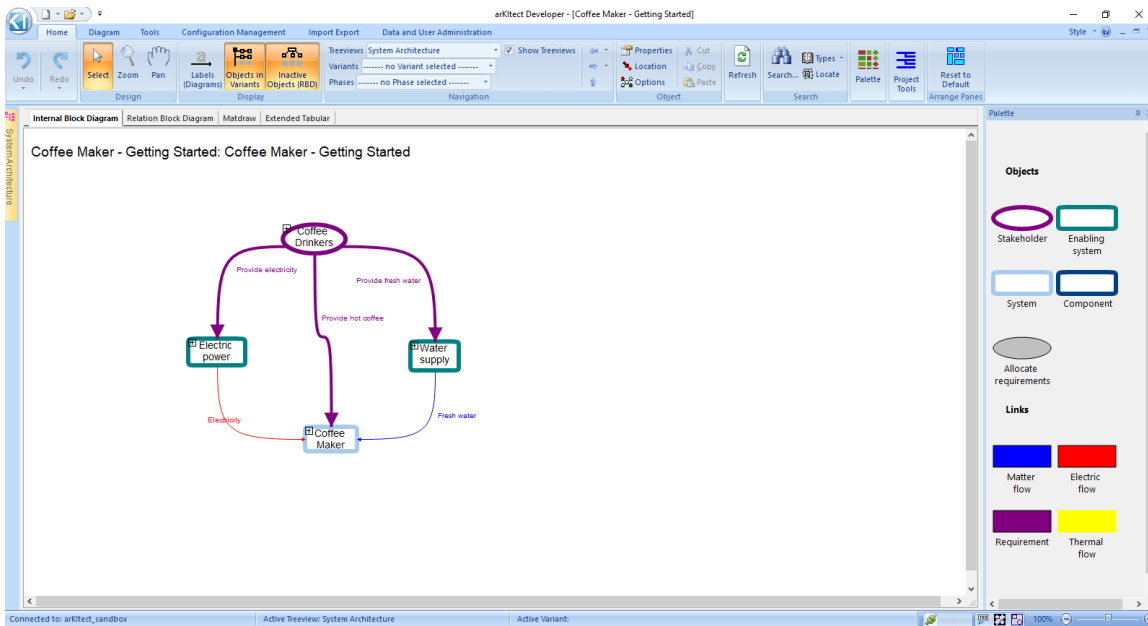
You can make a copy either in the same workspace or choose to export the copy to another workspace. Check the **Export to another workspace** check-box to copy the project to the **Sandbox** workspace. You will also need to login to this workspace. A default name is proposed for the copy; however, this name can be modified as wished.



The **Sandbox** workspace is intended to be used as a training environment; for your regular projects, we recommend that you use the **Default** workspace.

Now, a copy of the project has been created; however, the original project is still open (and we are still in the original workspace). Close the project via **Main button(File) Close**. You can now log on to the destination workspace by going to **Main button(File) Workspace Change Workspace** or simply using **Ctrl+W** shortcut. Open the copy via **Main button(File) Open** or simply using **Ctrl+O** shortcut.

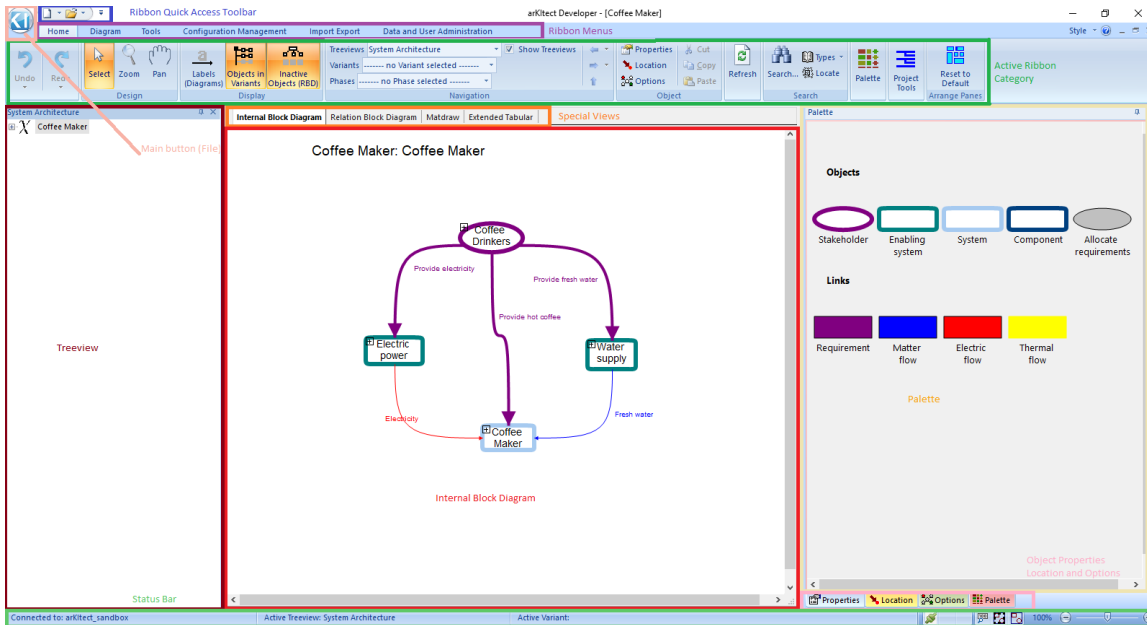
You should now have the following view in front of you:



The elements of this view are presented in detail in [Developer Environment](#).

Developer Environment

When a project is opened, the user is presented with the following type of view:



The following items can be identified:

- Main Button (File)
- Ribbon Menus
- Active Ribbon Category
- Ribbon Quick Access Toolbar
- Internal Block Diagram
- TreeView
- (Rules and Filters)
- Palette
- Status bar

Not all of these items are visible by default. Some of these items may be missing and others may be present depending on the last configuration employed by the user.

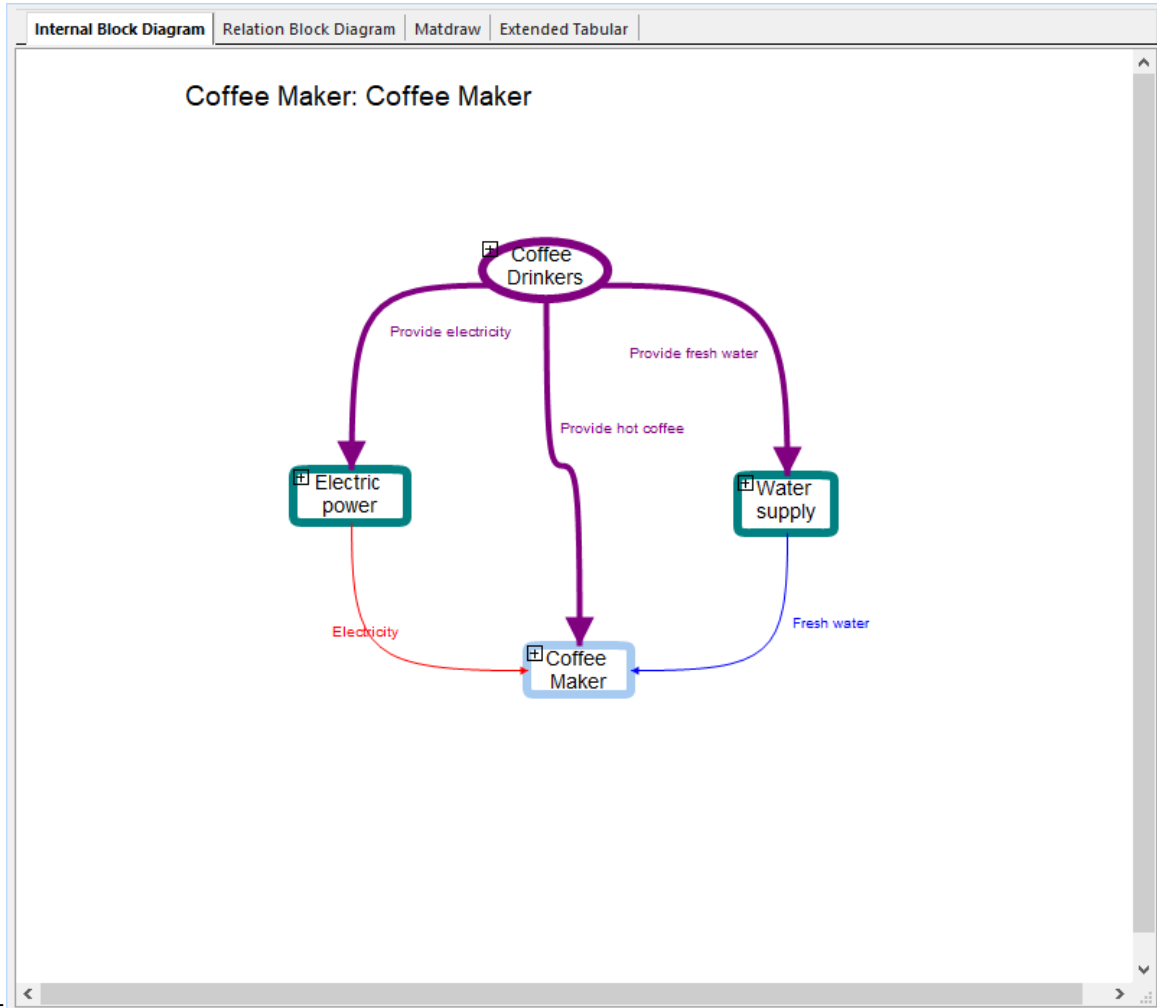
The **Main button (File)** serves as an entry point to main actions over projects (Open Project, Close Project, Copy Project, Change Workspace etc).

The **Ribbon menus (File (Main Button with arKitect logo), Home, Tools, Configuration Management, Import Export, Data and User Administration)** activates different **Ribbon Categories** which permit the user to access different project management and help tools. More information about each Ribbon Panel and Category can be found at [Ribbon](#) page.

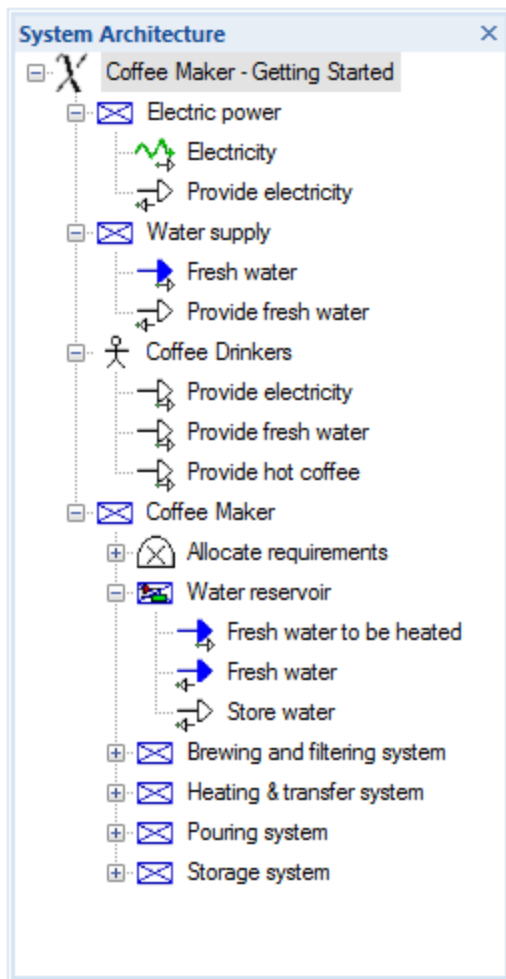
The **Active Ribbon Category** by default is always the Home Category.

The **Ribbon Quick Access Toolbar** is a very useful panel which allows user to configure the Ribbon so that main functionalities are placed there and are made always available without need to activate different Ribbon Categories. Also sometimes it allows even to minimize Ribbon thus saving more space for diagrams.

The **Internal Block Diagram** represents the graphical system representation. The system representation has different hierarchical levels; the navigation between these levels is explained shortly in [Navigation](#).



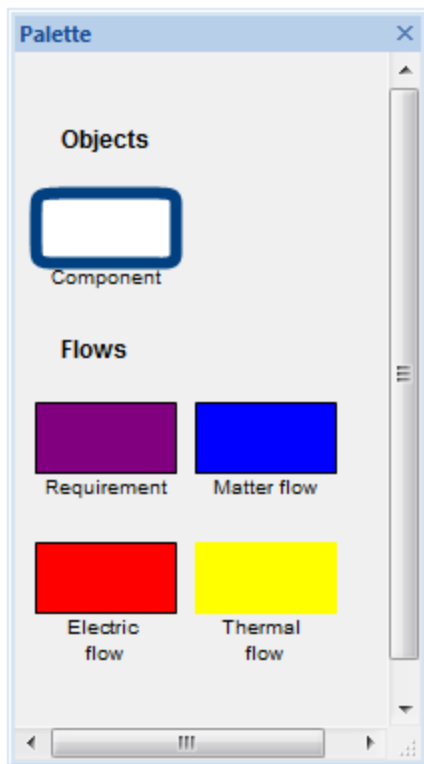
On the left-hand side of the window, the treeview can be accessed via a sliding panel. (The treeview is not displayed by default; make sure that **How Treeviews** button is checked in the [Navigation Panel of the Home Category](#) to make it visible.) The treeview provides a tree-form system representation. This representation corresponds to that of the Internal Block Diagram; these two tools are merely different means to visualize the same modeled system. These two tools permit in fact to visualize the object instances of the project.



The Rules and Filters panel on the left-hand side of the window is only visible for users with **arKitect Designer** rights.

The Palette at the right-hand side of the user interface is used to create objects (more information on this subject follows in [Working with Objects](#)).

The Palette permits to visualize the available types that can be added in the current Internal Block Diagram view.



It is possible to display several additional sliding and dockable panes. These panes include:

- Object Properties, Location and Options Window
- Variants and Options Manager
- Program Trace (for information on the execution of scripts)
- Project Tools (available scripts)

To display these panels, go to corresponding Category, locate a Panel with desired pane and activate the panes you need. It is possible to change the location of the panes and to dock and undock them as wished. The panes settings are saved and are therefore effective when you launch **arKitect** the next time.

At the bottom there is a Status Bar which gives user some information about :

- connection (workspace)
- active treeview and variant

It also allows to Zoom the current diagram using different buttons :

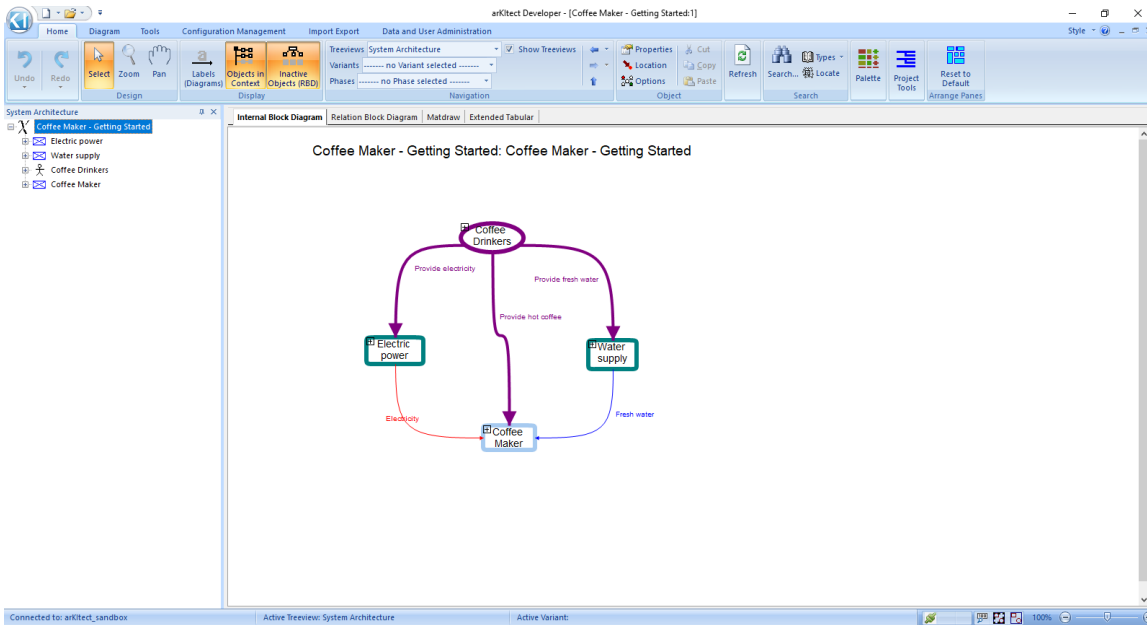
- apply 100% zoom
- zoom to selection
- zoom to fit
- define define specific zoom with -,+ buttons or slider

Now, go to [Navigation](#) to learn how to use the Ribbon Categories/Panels, the Internal Block Diagram and the treeview to navigate in the Coffee Maker - Getting Started project.

Navigation in the Diagram

Before starting to create and to manipulate objects in the Coffee Maker - Getting Started project, we shall first present the project and its structure briefly. The contents of the project can be discovered by navigating in the **Internal Block Diagram** and in the **Treeview**.

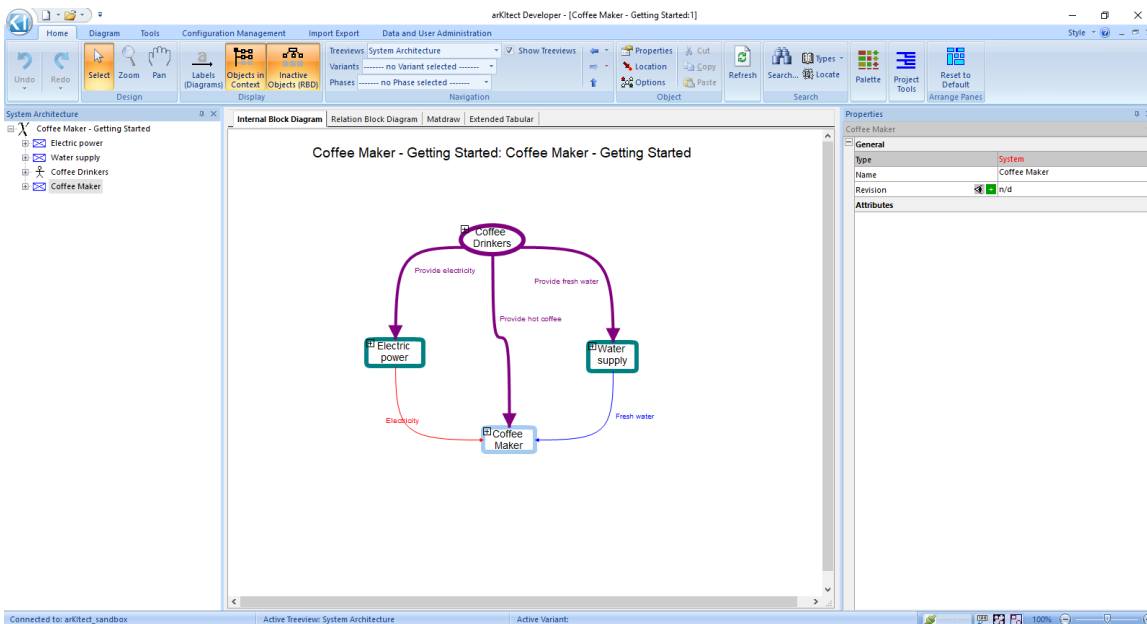
Start by docking the **Treeview** to the left-hand side of the user interface: in this manner, you can visualize both the **Internal Block Diagram** and the **Treeview** at all times. You can for the moment close the **Palette** panel (via the cross in its top right-hand corner). This provides more space for the **Internal Block Diagram** visualization.



Let us start first to explore the project via the **Internal Block Diagram**. In this diagram, we can see the several objects, namely

- Coffee Drinkers
- Coffee Maker
- Water Supply

These objects are actually object instances: they each have a unique name and a situated at a given location in the object instance hierarchy. The types corresponding to these object instances can be seen presented in the **Properties** panel at the right-hand side. Display the **Object Properties** panel through the **Object Panel** of the **Home Category** (**Properties** button) or by using **Ctrl+Shift+P** shortcut. Click (a single left-click) on objects to select them.

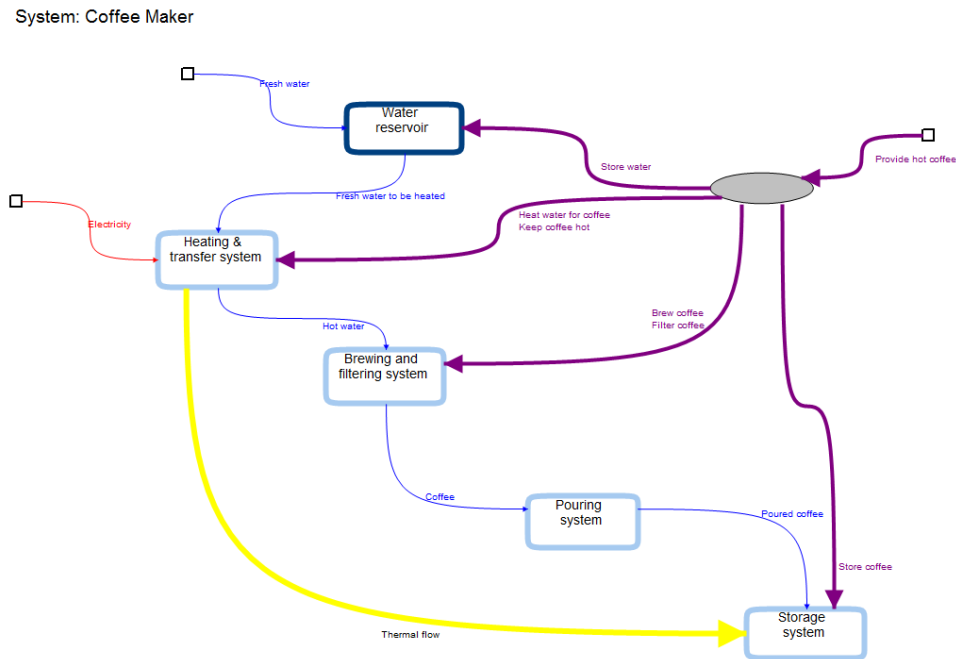


The types represent an abstract "mold" used to create object instances; the type defines the characteristics of the object instances, such as their appearance, the possible locations in the object hierarchy, available child types, etc.

For instance, the object named *Coffee Drinkers* is of type *Stakeholder* while the *Coffee Maker* and *Water Supply* are of type *System*. We can also see several objects represented as flows instead of containers in the **Internal Block Diagram**. For instance, the *Coffee Drinkers* object produces a violet flow of type *Requirements* consumed in the *Coffee Maker*.

Flows are also objects and the same rules apply to them as to objects represented in container-form. In fact, the starting and end points of flows (i.e. the locations where they are produced and consumed) are represented as containers.

To explore the inner structure of the *Coffee Maker* object, double-click on it in the **Internal Block Diagram**. You are now presented with the following view:



We are now viewing the internal structure of the *Coffee Maker* object. We can see that the name of the parent object (*Coffee Maker*) is actually displayed at the top left-hand corner of the Internal Block Diagram. The *Coffee Maker* contains several subsystems, notably

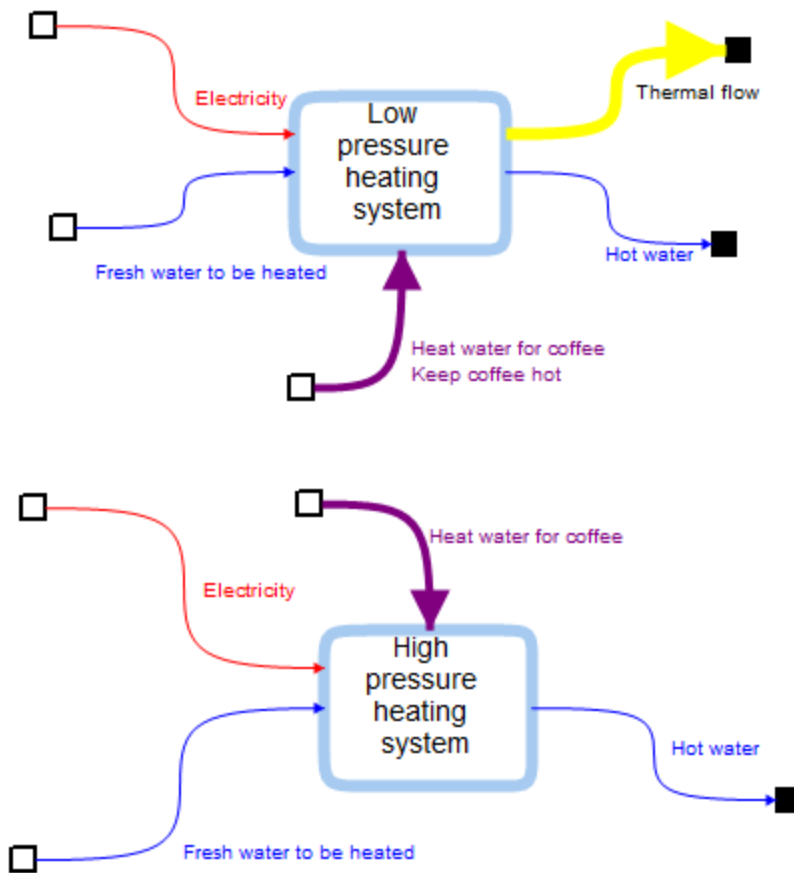
- Water reservoir
- Heating & transfer system
- Brewing & filtering system
- Pouring system
- Storage system

These subsystems are first-degree child objects of the *Coffee Maker*; *Coffee Maker* constitutes the parent object of these object instances.

The *Water Reservoir* is actually of type Component while all the other listed elements are of type System. As the *Coffee Maker* itself is of type System, this type is of recursive nature. We can also see flows connecting the different subsystems as well as flows and an object related to the stakeholder requirements.

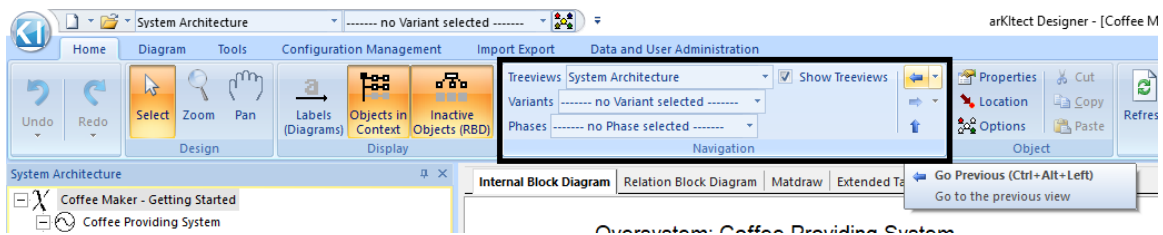
Now, double-click on the *Heating & transfer system* object. You can see that there are even more subsystems inside this object, notably *Low pressure heating system* and *High pressure heating system*.

System: Heating & transfer system

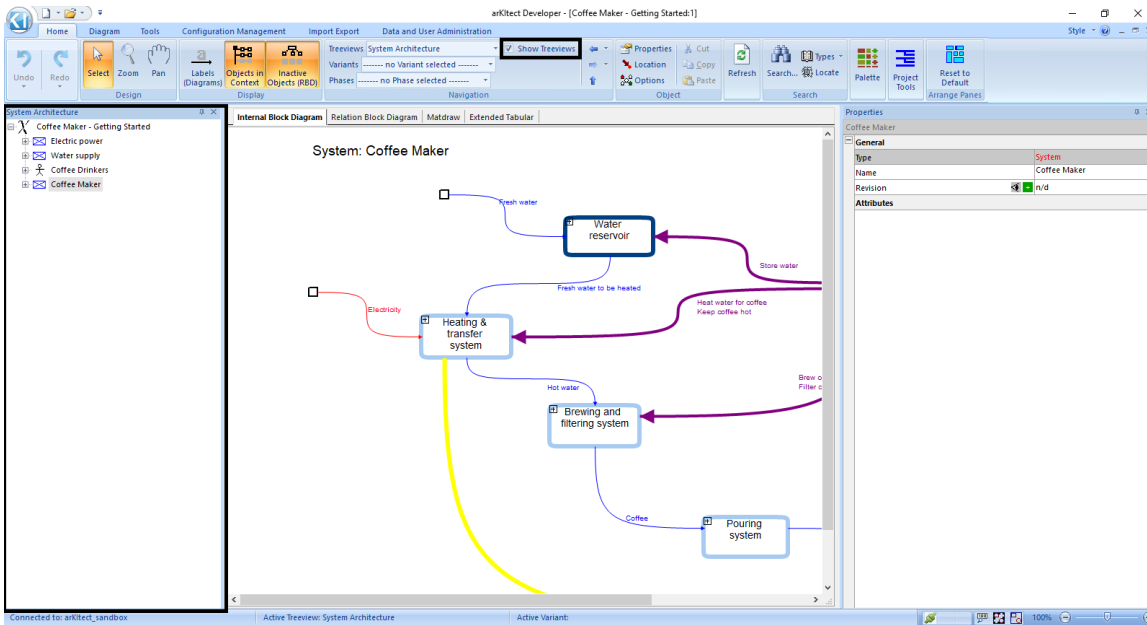


The flows originating from outside the focus system (*Heating & transfer system*) are represented using a white port while flows consumed outside the focus system end in a black port.

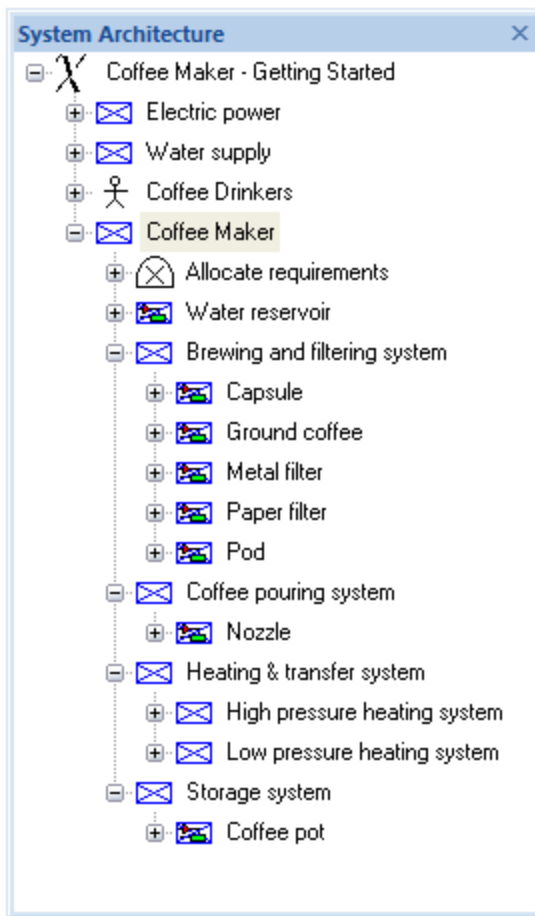
Let us now go back up in the object hierarchy by using the **Navigation Panel** of the **Home Category**. The **Go Previous** button allows you to go back to the previous focus object. Press **Go Previous** (or use **Ctrl+Shift+Left** shortcut) and you return to the *Coffee Maker* system. The **Go Next** button allows you to go forward in the object hierarchy (pressing **Go Next** (**Ctrl+Shift+Right** shortcut) returns you now to the *Heating & transfer system*). The **Go Parent** button takes you one hierarchical level higher (**Ctrl+Shift+Up** shortcut).



You can also use the treeview to navigate in the object hierarchy. If the treeview panel is not displayed, click on the **Show Treeviews** button to display it.



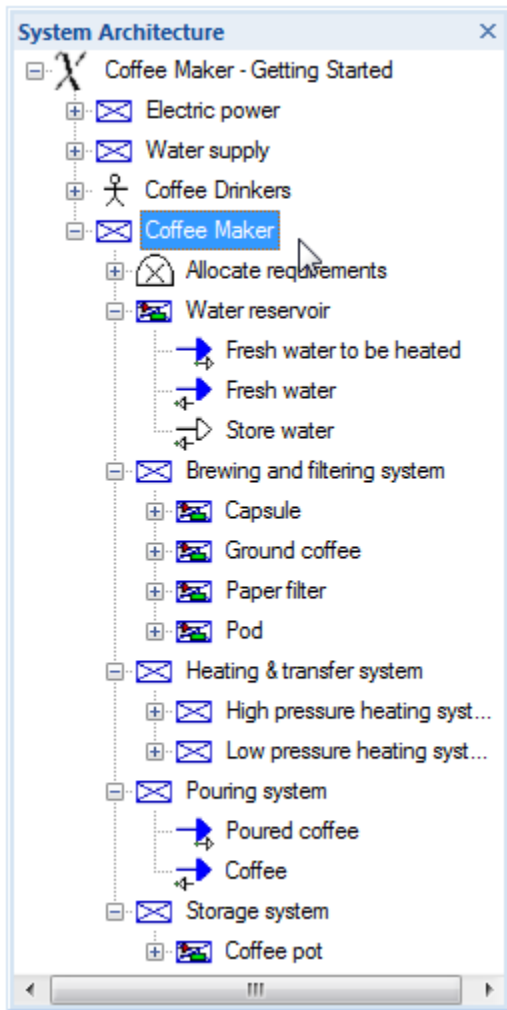
At the moment, you should find yourself at the *Coffee Maker* system. Take a look at the **Treewiew**, you should see that the *Coffee Maker* is highlighted in light gray.



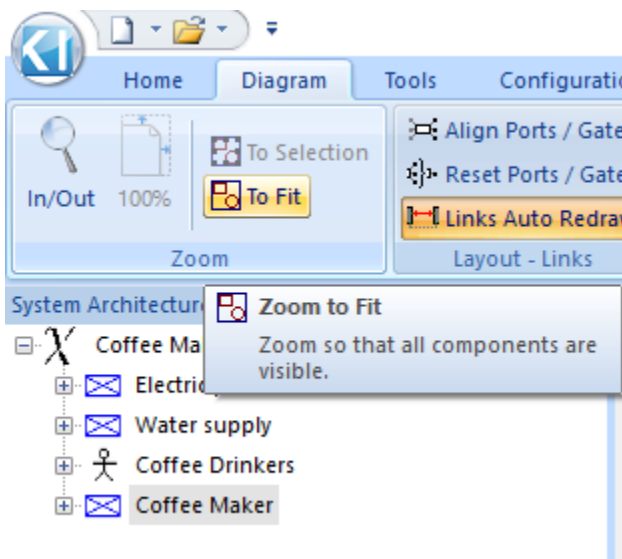
If you click on the *Brewing and filtering system* in the treeview, the view in the **Internal Block Diagram** changes: focus is now on the *Brewing and filtering system*. Also, in the **Treewiew**, this system is now highlighted. If you now click on the **Go Parent** button (or use **Ctrl+Shift+Up** shortcut) in the Navigation Panel of the Home Category, you return to the *Coffee Maker* level. Note that the *Coffee Maker* becomes highlighted also in the **Treewiew**.

These two different representations of the system, the **Internal Block Diagram** and the **Treewiew** are linked. They are also equivalent in the

sense that they both contain all the same objects. You can verify this by opening the nodes of the **Treeview** and exploring their contents. The flows can be seen under the nodes of the final objects producing or consuming them.

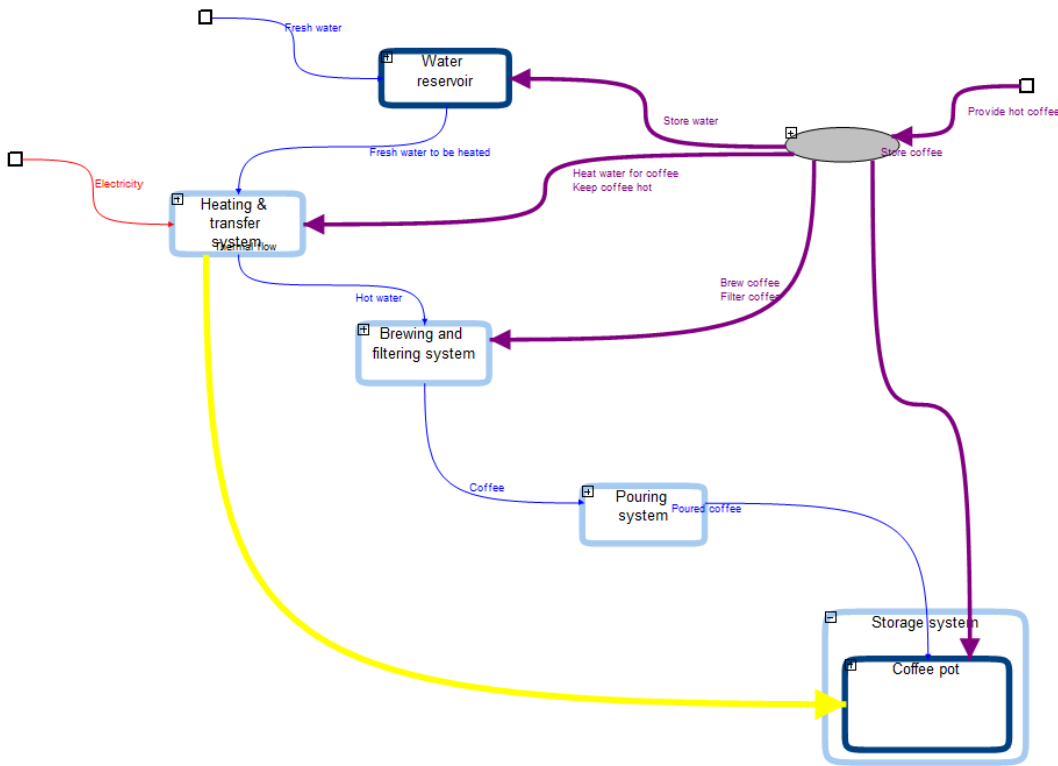


There are also a few other useful navigation tools, like **Design Panel of the Home Category** with its **Zoom** and **Pan** buttons or **Zoom Panel of the Diagram Category**. You can for instance **Zoom** in on the **Internal Block Diagram** or, if you want to display all the diagram objects at a time, you can use the **Zoom to fit** tool.



Return to the *Coffee Maker* object. Sometimes one wishes to display the internal structure of a given object in the **Internal Block Diagram**. To do this, right-click on the chosen non-flow object, the *Storage system*, and choose **Expand** from the context menu. This enlarges the *Storage system* object and displays the *Coffee pot* component inside it. The same can be achieved if double-clicking on [+] button near the *Storage system*,

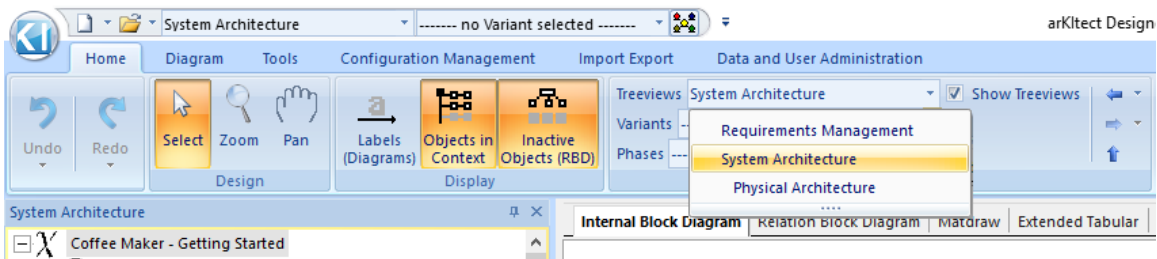
System: Coffee Maker



Next, we learn about Navigation in Projections.

Navigation in Projections

You now know how to navigate in the **Internal Block Diagram** and in the **Treeview**. However, **arKitect** is a multi-view visualization tool and it is thus possible to have several different views on the system. These views, they are actually called **projections** in **arKitect** terminology, are accessed via the **Navigation Panel** of the **Home Category**. Projections are created by applying corresponding filters which permit to filter out certain parts of the project.

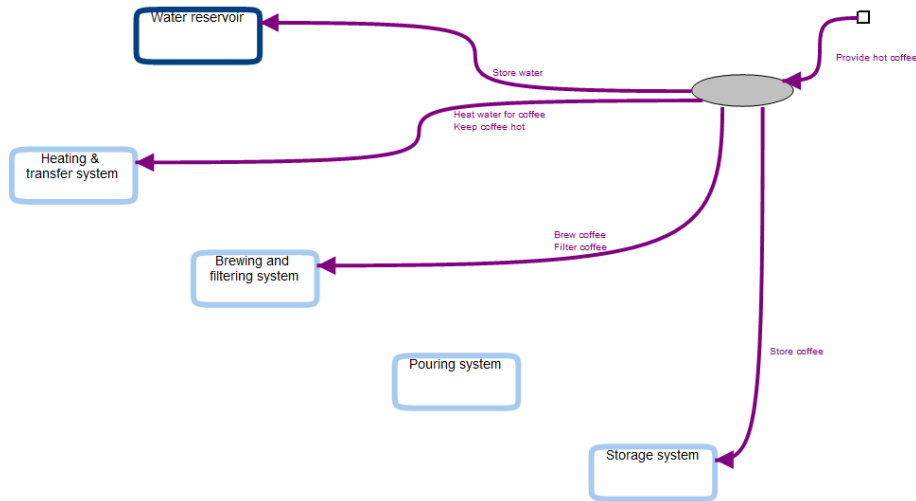


We can see that the currently active projection is called *System Architecture*; if you click on the arrowhead next to the projection name, you can see that there are in total three projections, namely:

- Requirements Management
- System Architecture
- Physical Architecture

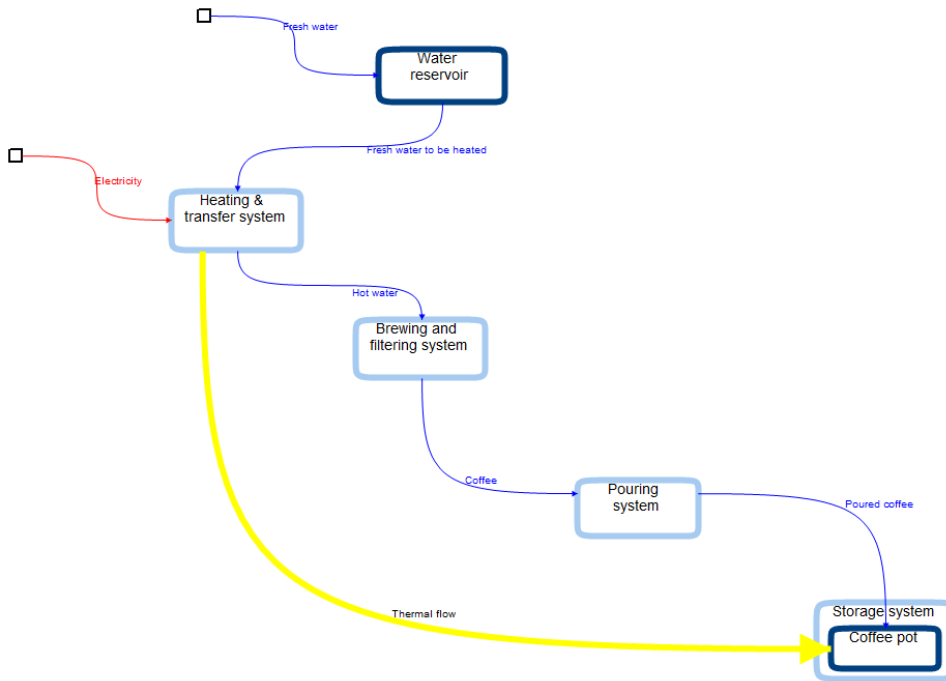
Select *Requirements Management* from the **Treeviews** list; this activates the *Requirements Management* filter and displays the objects visible in this projection. In the **Internal Block Diagram**, double-click on the *Coffee Maker* object. You can now see that this view resembles *System Architecture*; however, only the stakeholder *Requirements* and *Systems* are displayed.

System: Coffee Maker



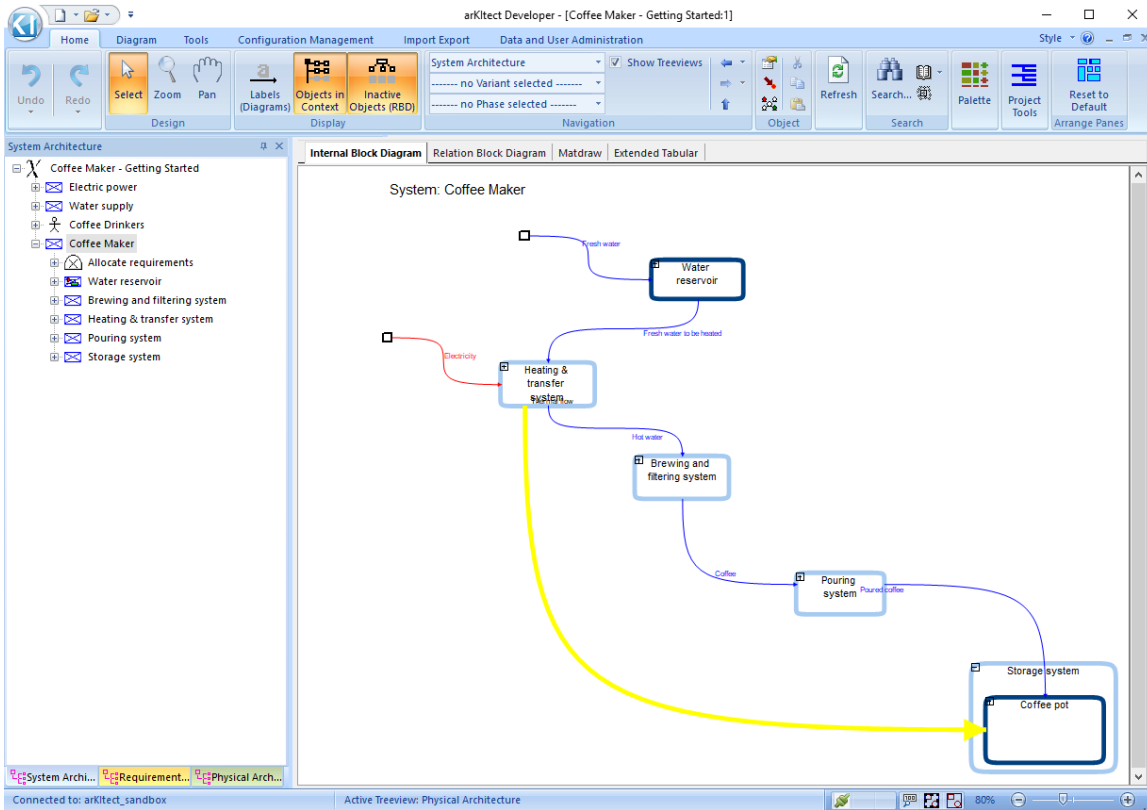
Activate now the *Physical Architecture* filter, and double-click on the *Coffee Maker* object in the **Internal Block Diagram**. The *Requirement* flows have now disappeared and only the *Systems*, *Enabling Systems*, *Components* and the *Thermal*, *Electrical* and *Matter* flows are visible. In addition, the *Liquid Type* attribute of Matter flows is no longer visible on the flow containers.

System: Coffee Maker



You can see that the *Storage system* object is expanded (this action was taken in the *System Architecture* projection). The *Physical Architecture* filter is actually a subfilter, meaning that it is linked to the *System Architecture* filter. A subtree can only contain object types that are also present in the parent filter and the object positions between the two projections are linked: moving an object in the **Internal Block Diagram** of *Physical Architecture* also moves it in the *System Architecture* projection and vice versa.

As you have now opened all three projections, you can see them appear as tabs visible at the left-hand side (if the **Treeview** panel is not displayed, click on the **Show Treeviews** button in the **Navigation Panel of the Home Category**). You can use these tabs to navigate between projections.



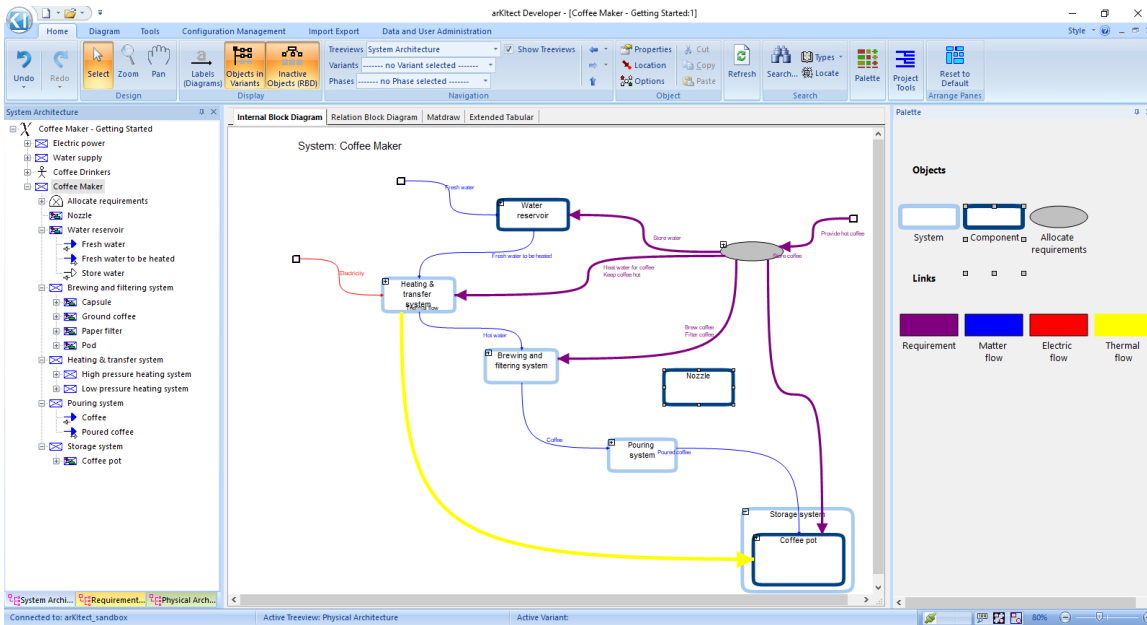
The *System Architecture* filter actually contains all the object types existing in the project while the other two filters only display certain chosen types.

Now that you are familiar with the navigation aspects, we can start [Working with Objects](#).

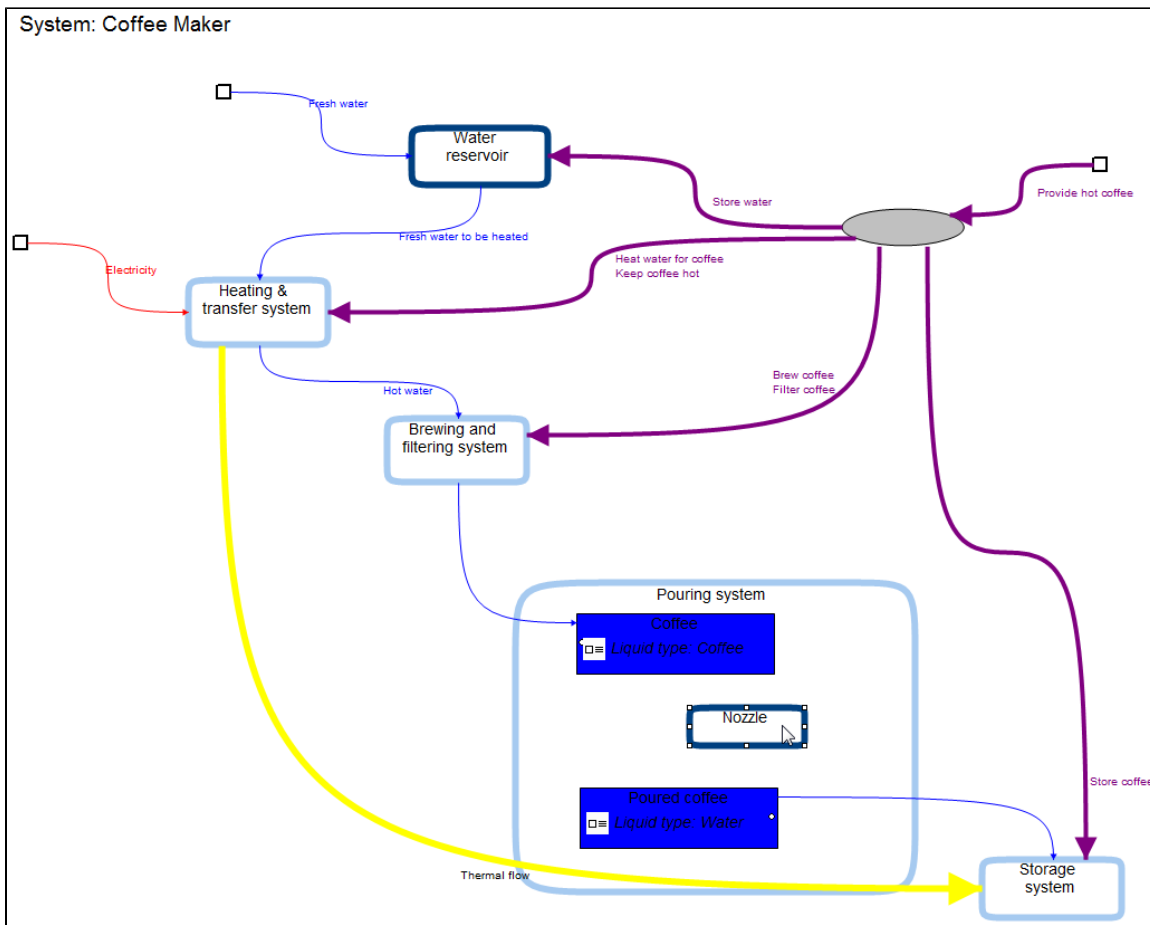
Working with Objects

Return to the System Architecture projection. Select the *Coffee Maker* object and double-click on it in the **Internal Block Diagram** to focus on it.

We want to add an object instance using the **Palette**. (If the **Palette** is not visible, activate it at [Palette Panel of the Home Category](#)) Drag an object of type Component from the **Palette** and drop it somewhere on the **Internal Block Diagram** background. (This action corresponds to the **Add New Object** command presented later.) Name the object *Nozzle* and press ENTER. You can resize the object by clicking on it and dragging its outline to the wanted size.



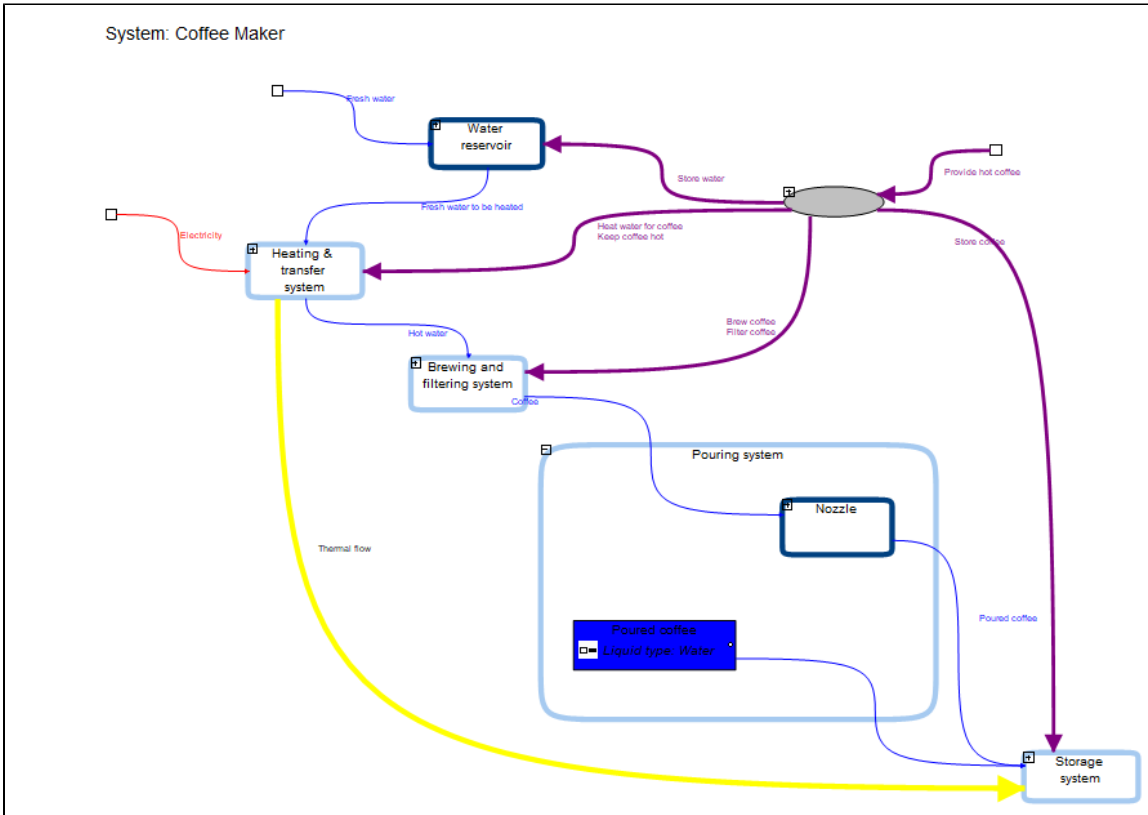
However, the *Nozzle* object is now a first-degree child of *Coffee Maker*, we actually wish to place this object as a child of the *Pouring system*. In order to do this, **Expand** the *Pouring system* object and do a right-click drag and drop to transfer *Nozzle* inside it. Select **Move here** from the appearing context menu. A new object now appears inside the *Pouring system*.



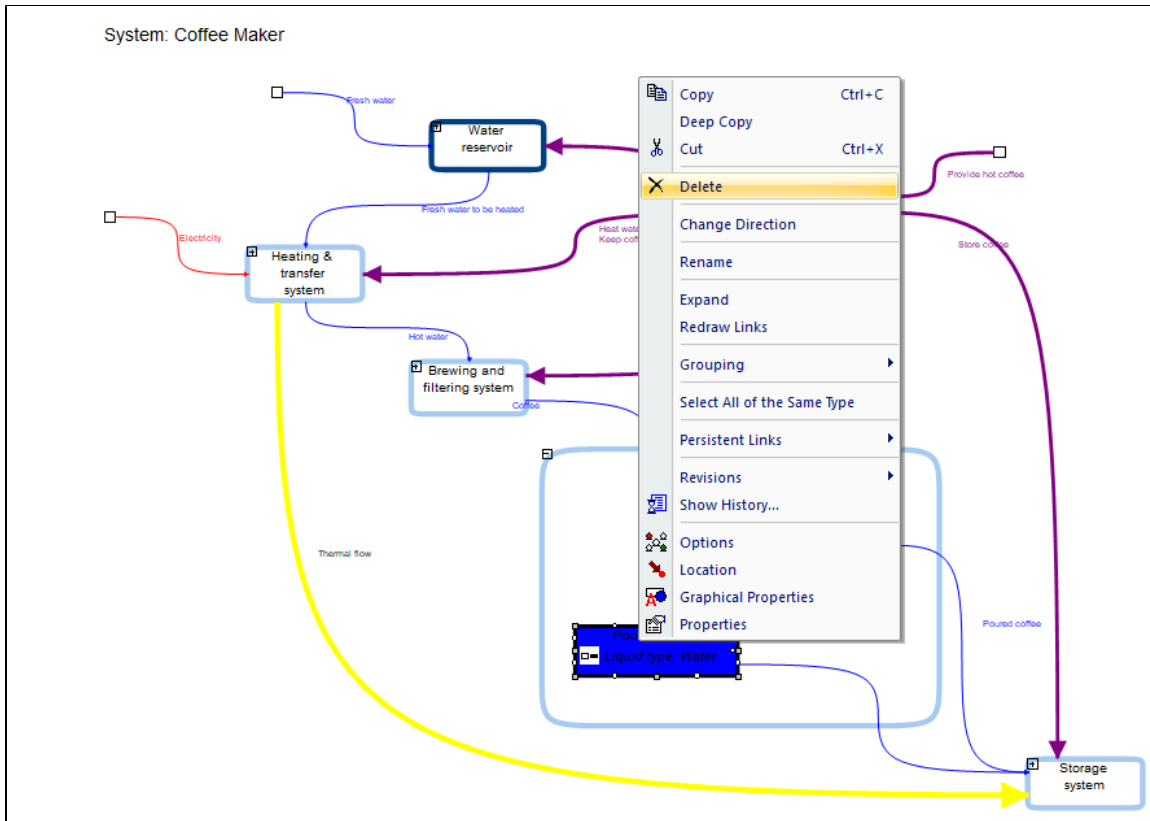
We also wish to allocate two flow objects consumed and produced inside the *Pouring system* to this newly created *Nozzle* Component. These flows are actually represented with flow containers inside the *Pouring system* object. A flow container symbolizes the fact that a flow begins (is produced) or ends (is consumed) inside a given system.

To allocate the *Coffee* flow, click on its flow container and drag and drop it with a left-click to the *Nozzle* Component. Choose **Move here** from the

context menu. Now, drag and drop *Poured coffee* with a left-click and choose **Copy as Output** in the context menu. If you now double-click on the *Nozzle*, you can see that the flow containers are inside this object, meaning that it produces and consumes the corresponding flows. This can also be seen in the treeview by displaying the *Nozzle* node.

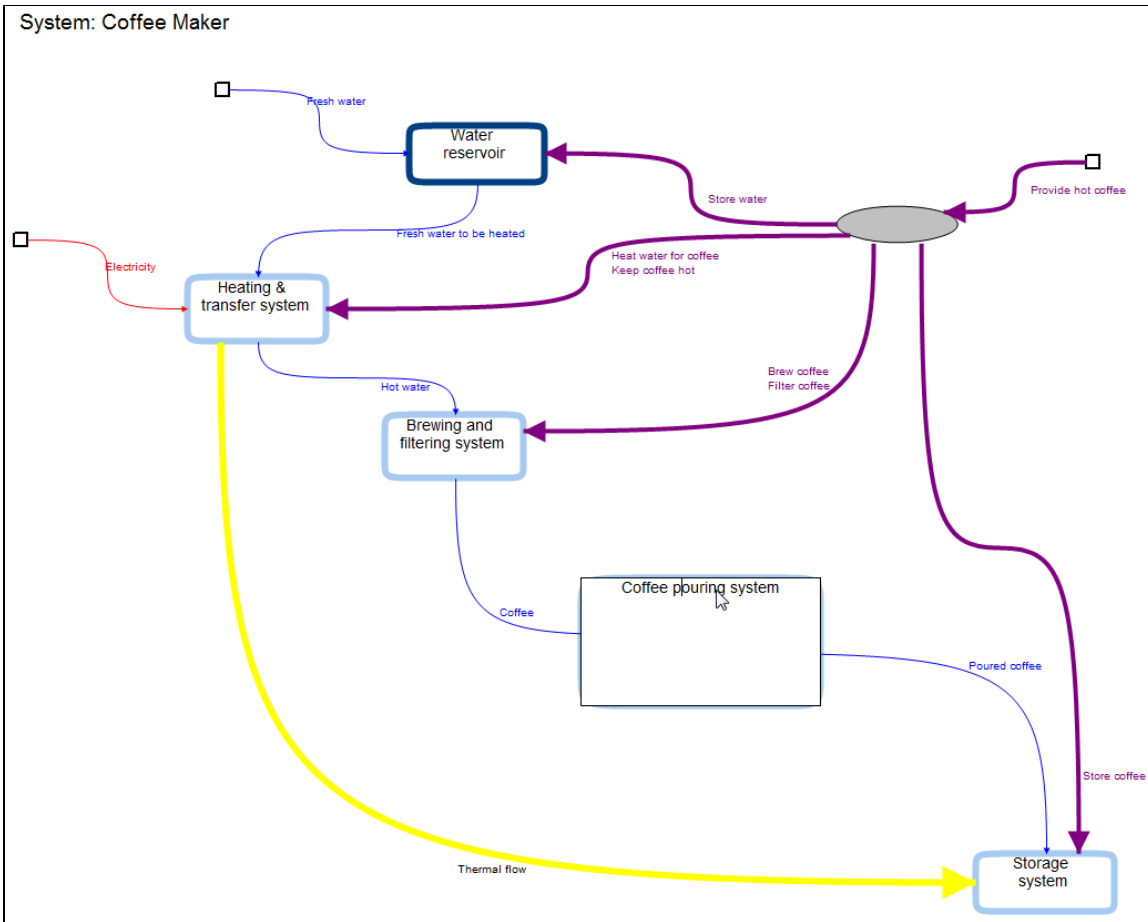


You can see that the left-click dragging and dropping of objects with **Move here** transfers them to the new parent object while a left-click drag and drop and **Copy Here** duplicates the reference to the object. (Left-click drag and drop and **Copy Here** corresponds to the **Add Existing Object** command presented below.) *Poured coffee* is thus now produced in both the *Pouring system* and in its *Nozzle* component. As this is not very realistic, let us delete the flow from the *Pouring system*. Right-click on the flow container inside the *Pouring system* object and choose **Delete** from the context menu (NB! use **Delete Locally** in the confirmation dialog to remove only the reference and not the object itself).



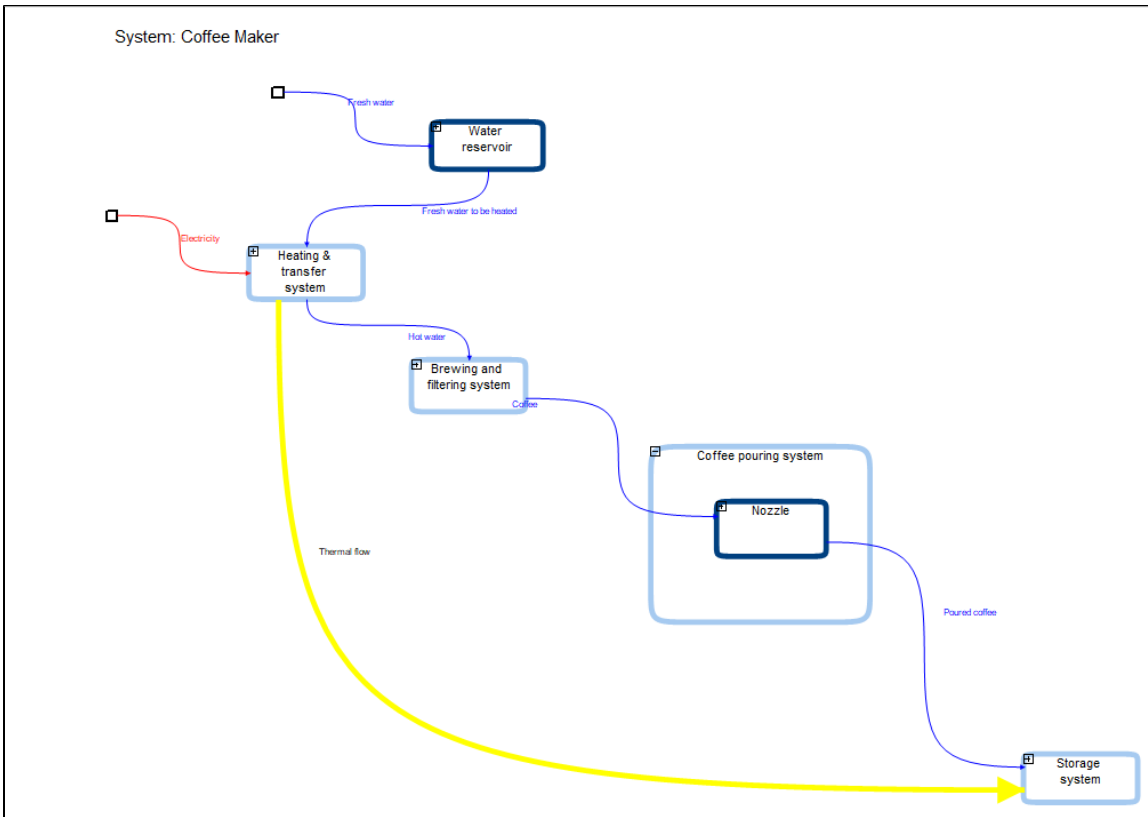
You can now resize the *Pouring system* container to better fit the size of *Nozzle*.

If you have made a mistake in naming an object and you want to change the name later, just right-click on it and choose **Rename** from the context menu. Let us now rename the *Pouring system* as *Coffee pouring system*.

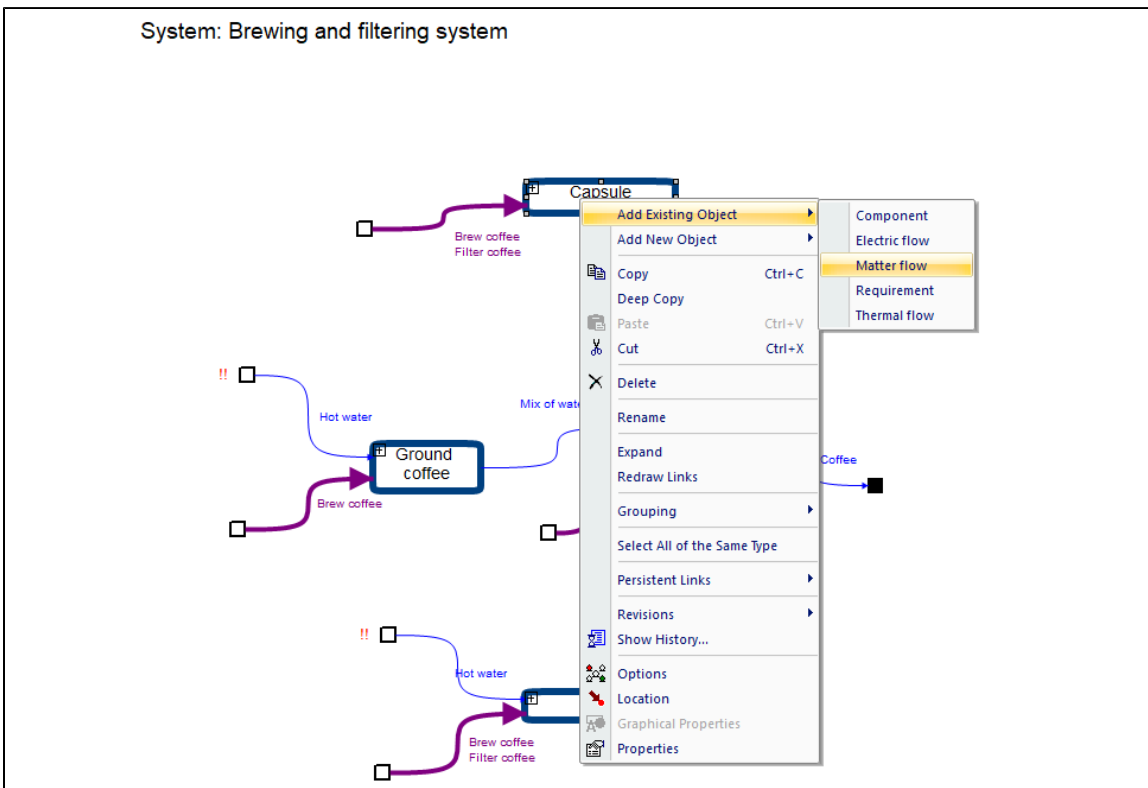


It is not possible to add objects from the **Palette** everywhere in the **Internal Block Diagram**; for instance, an object of type *System* cannot be added inside a *Component*. The objects types have a predefined hierarchy that determines whether an object instance can be created at a given hierarchical location.

Now, go to the *Physical Architecture* projection. You can see that the *Nozzle* Component is visible inside the *Coffee pouring system*: the changes made in one projection thus also affect other projections. This is because the *Nozzle* object visible in the two projection is actually the same object instance. As their name indicates, filters merely filter the project and render given object types visible or invisible in the created projections; in the *Physical Architecture* projections, the Stakeholders and Requirements have been filtered out.

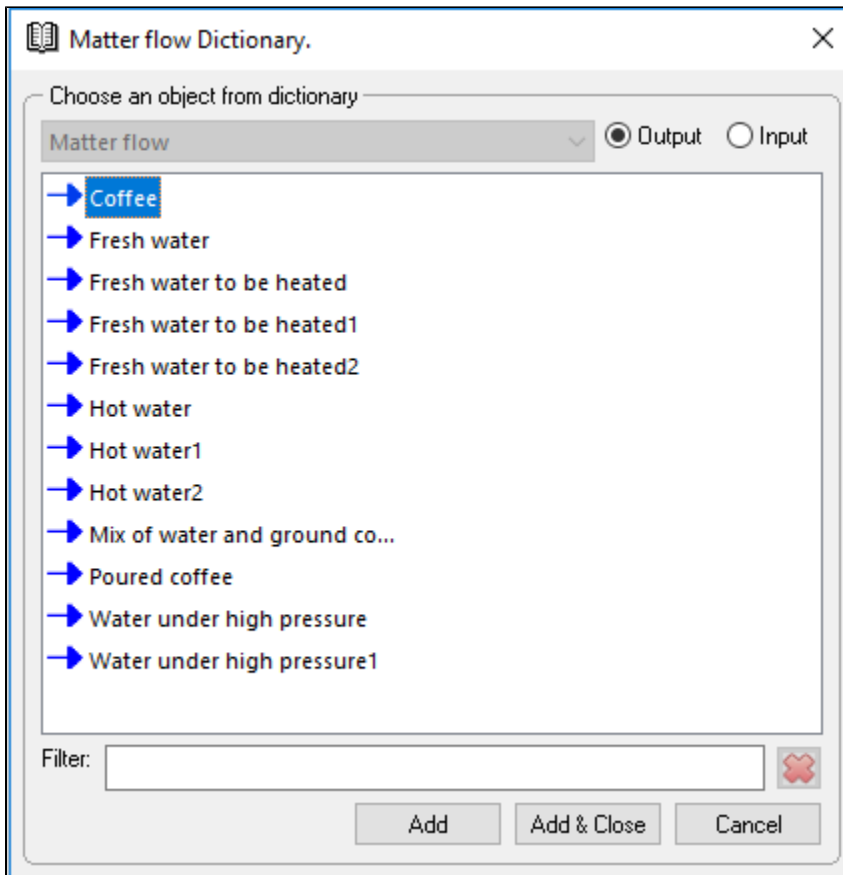


Return to the System Architecture projection and go inside the *Brewing and filtering system*. You can see that the *Capsule* Component receives no Matter flows while for instance the *Pod* Component consumes *Hot water* and produces *Coffee*. We now want to implement the same for *Capsule*. As these two flows already exist, we only need to add references to them (*Hot water* is produced somewhere inside the *Heating & transfer system* while the *Coffee* flow is consumed in the *Nozzle* Component of *Coffee pouring system*). To add a reference to an existing object, right-click on *Capsule* and choose **Add Existing Object** from the context menu. You are now presented with a list of available types.



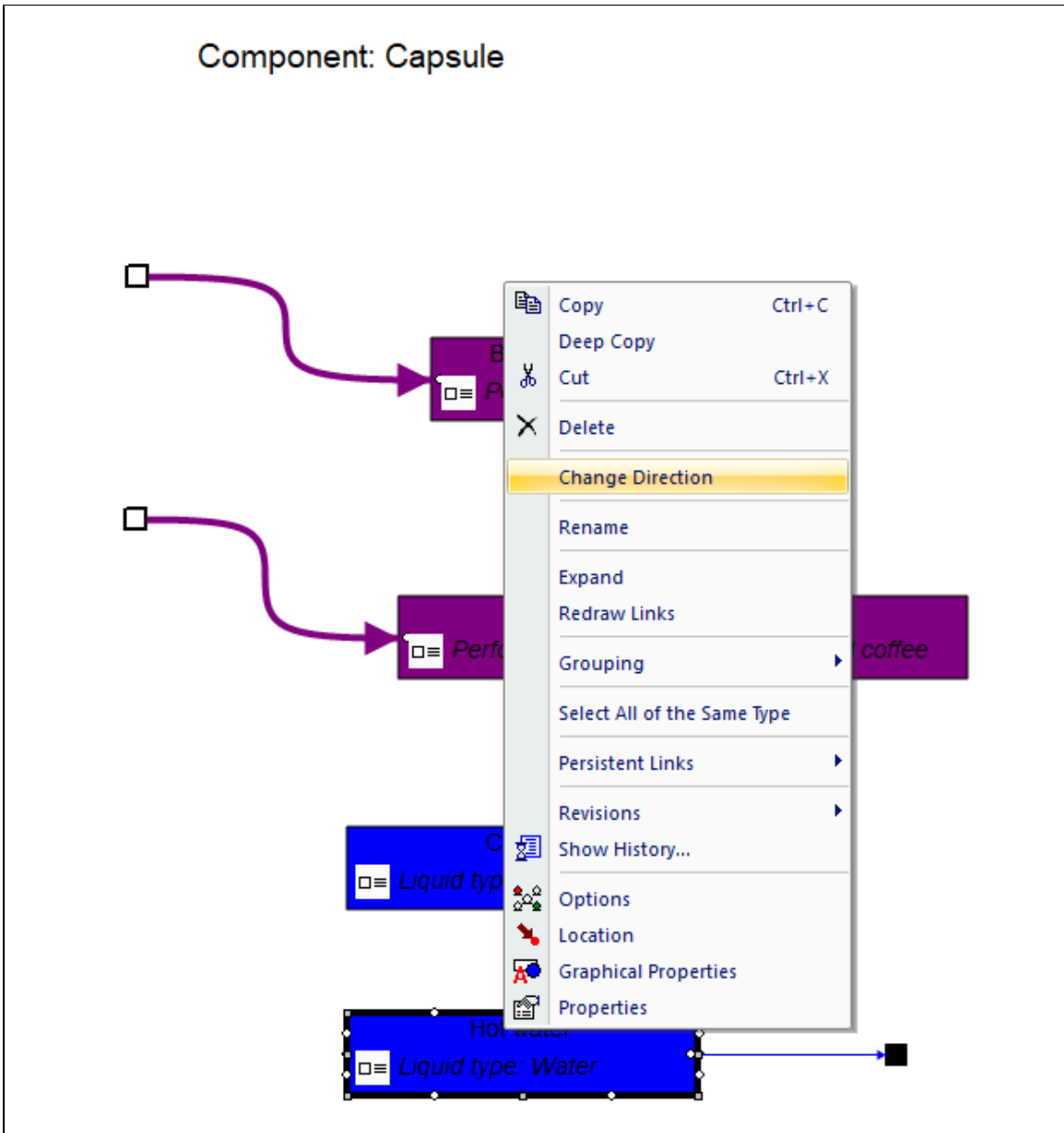
Choose the Matter flow type from the list. You are now presented with the Matter flow **Dictionary** which displays all the object of the Matter flow

type.



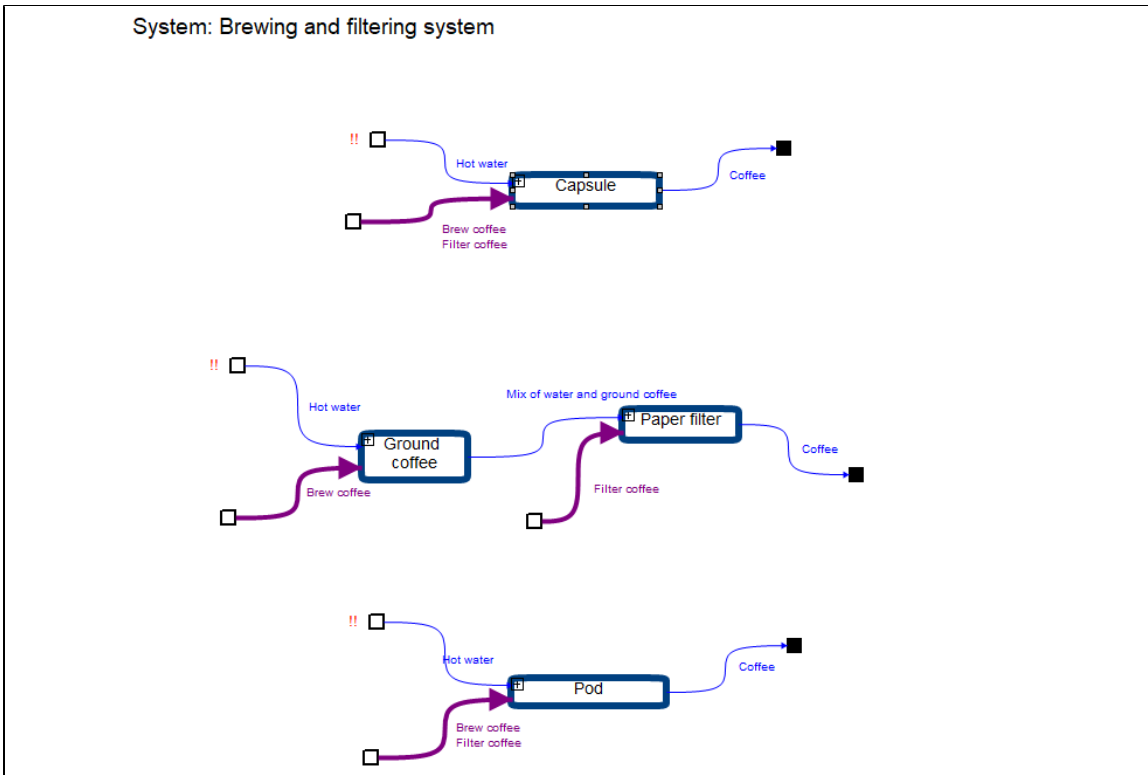
Choose *Coffee* from the list and click **Add**. This adds a new reference to the *Coffee* flow. However, we also want to add a reference to *Hot water*. Choose *Hot water* from the list and click **Add** and then **Close**. The new flows now appear for *Capsule*. However, we can see that there is a mistake: the *Hot water* flow should be consumed by the container and not produced by it. This could have been specified in the Dictionary window where the **Produced** option is proposed by default.

In order to rectify the situation, double-click on *Capsule* and go to the *Hot water* flow container. Right-click on the flow container and choose **Change Direction** from the context menu. You can now go up to the *Brewing and filtering system* level and verify that all is ok.



If both the flow producer and consumer have been defined wrongly, it is necessary to **Change Direction** twice: once at the starting point of the flow and once at its endpoint. However, in our case, there was only a mistake at the consumer end (which had been defined as producer).

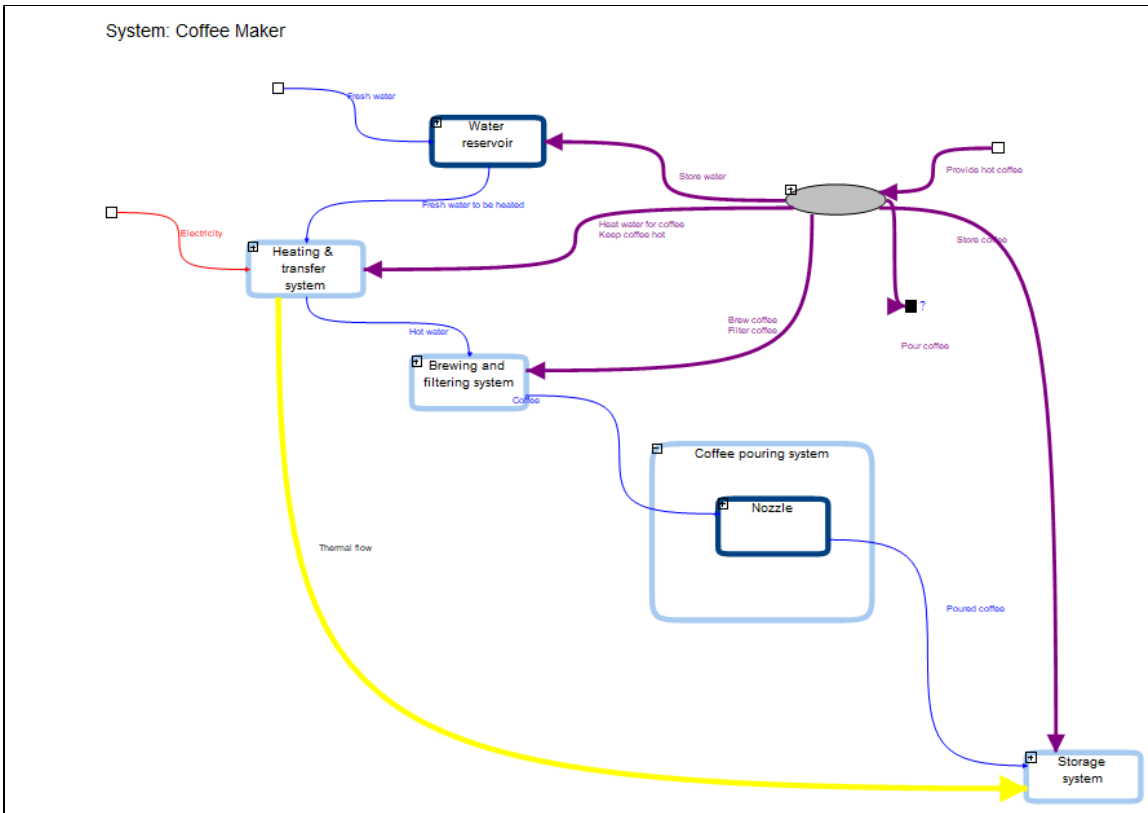
There are now several references to the *Hot water* and *Coffee* flows; there are at once three different consumers and two possible producers. The existence of multiple producers is marked with a red double exclamation mark.



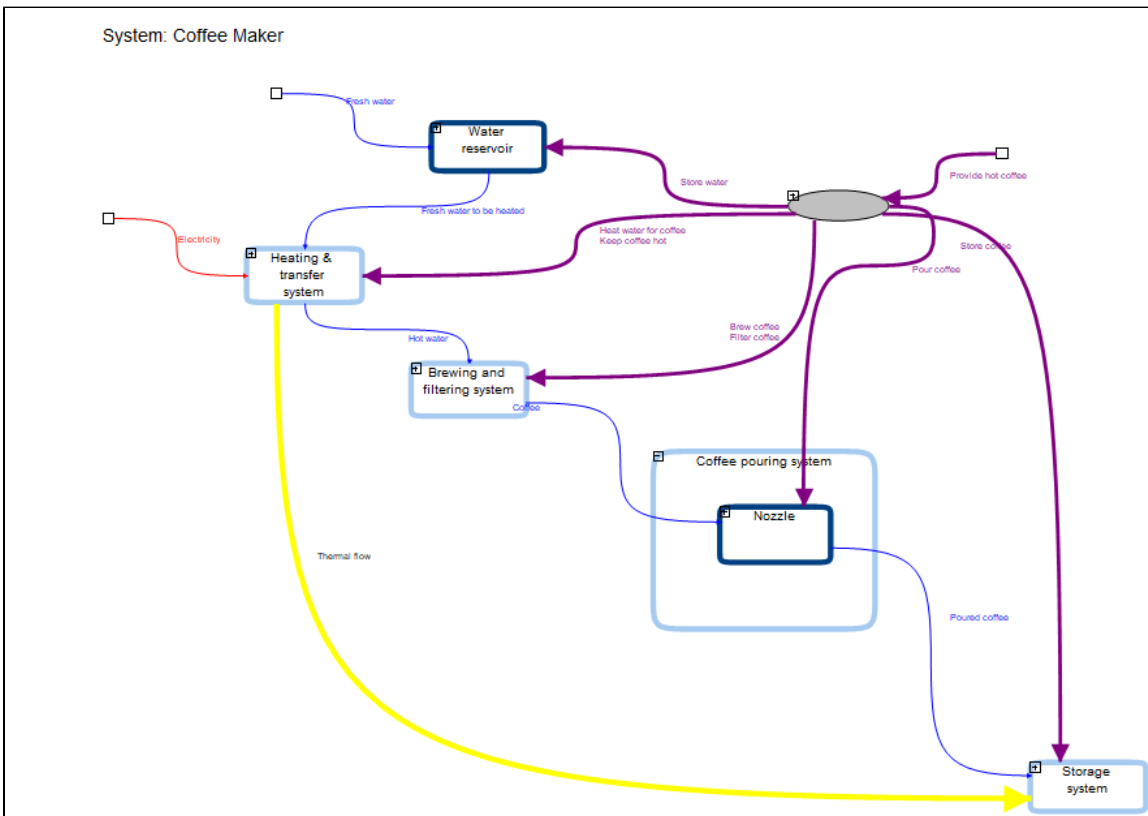
It is also possible to create multiple references to non-flow objects; for instance, the functions of a system can be modeled both in functional architecture and in system architecture, allocated to system components. In this case, filters can be used to create projections on these two different system aspects, functional architecture and system architecture. The functions seen in functional architecture et system architecture projections are now different object instances or references.

It is not possible to add references to existing objects everywhere in the object hierarchy: a new reference can only be created at a different location in the object hierarchy. Go back up in the hierarchy so that *Coffee Maker* is again the focus object. Right-click on the Internal Block Diagram background choosing **Add Existing Object** and the Component type. Then choose *Water reservoir* from the Component Dictionary, click **Add** and then **Close**. However, no new reference to *Water reservoir* appears on screen: this is because the *Water reservoir* already exists as child object of the *Coffee Maker* and it cannot thus be present twice at the same location.

Now we wish to add a new requirement consumed by the *Nozzle* Component. Right-click on the gray How to object that interprets higher-level stakeholder requirements in terms of technical requirements. Choose **Add New Object** and then the Requirement type as Output from the context menu. Name the new requirement *Pour coffee*. You see that a new requirement flow now appears; however, the flow ends in a question mark. This signifies that the consumer of the flow has not yet been defined. In general, unallocated flows (flows with either no consumer or no producer) are marked with a blue question mark.

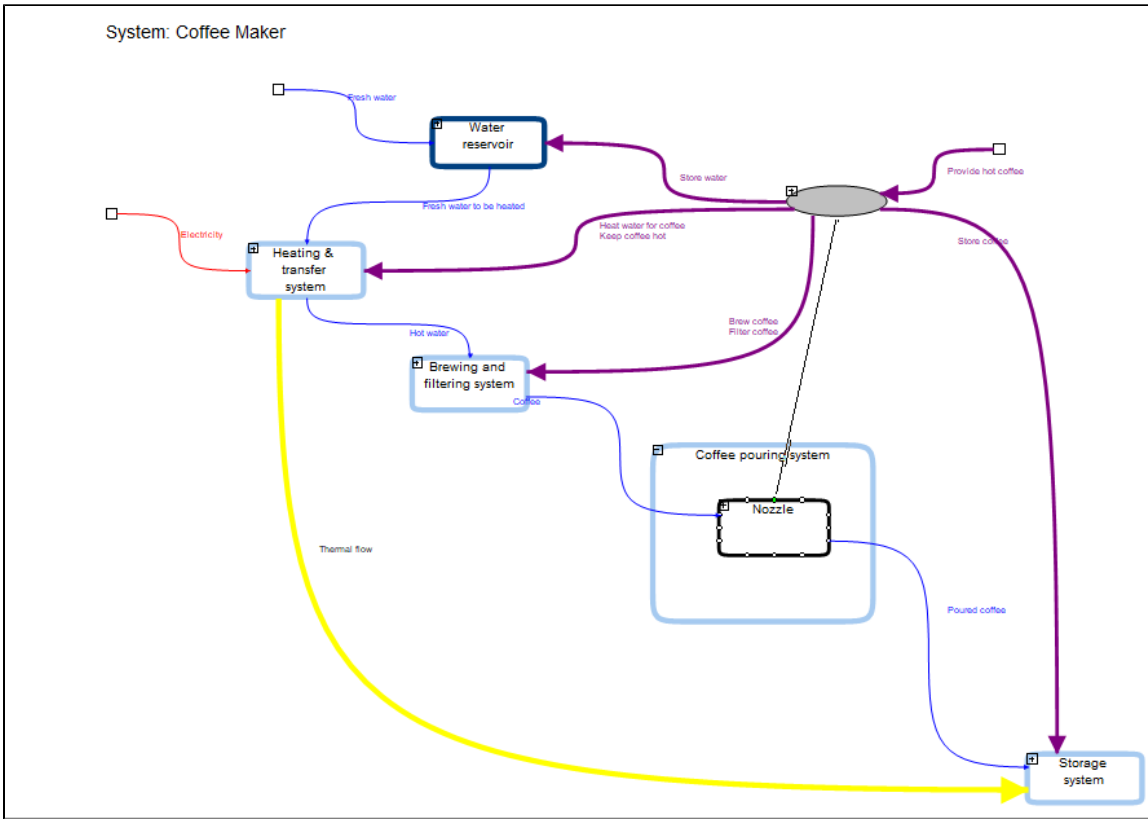


In order to allocate the *Pour coffee* flow to the *Nozzle* Component, drag and drop the black flow port to *Nozzle* with a left-click and select **Copy as Input** from context menu. You now have the following view:

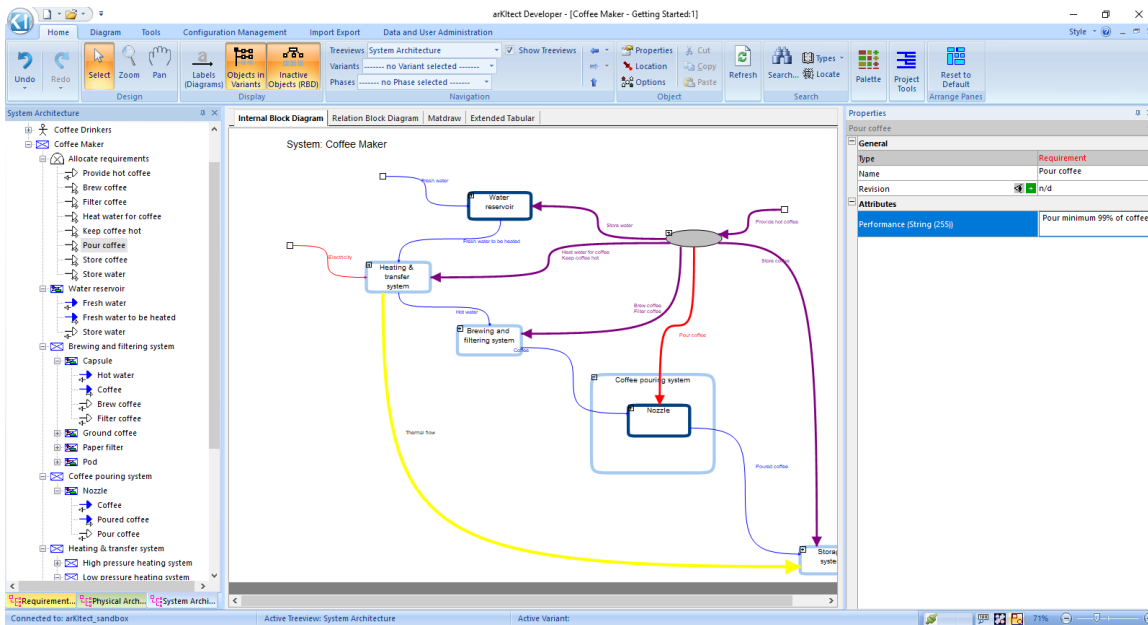


New flows can also be created graphically in the Internal Block Diagram. Delete the *Pour coffee* requirement: right-click on the flow and choose **Delete From Both Ends** from the context menu, then confirm that delete will be done globally. This deletes the flow both from the producer and the consumer end. As all references to the *Pour coffee* flow have now been deleted, it is no longer possible to employ **Add Existing Object**.

In order to create the flow anew go to the bottom part of *How to object*. Trace the cursor over the border of the container until it becomes outlined and small dots appear. Click once on a green dot and bring the cursor to the border of the *Coffee pouring system* container. Here again trace the cursor over the border of the container until it becomes outlined and small dots appear. Click again when the green dot indicates a suitable spot. In the dialog that appears name the new requirement *Pour coffee*. A Requirement flow should now appear linking the two containers.

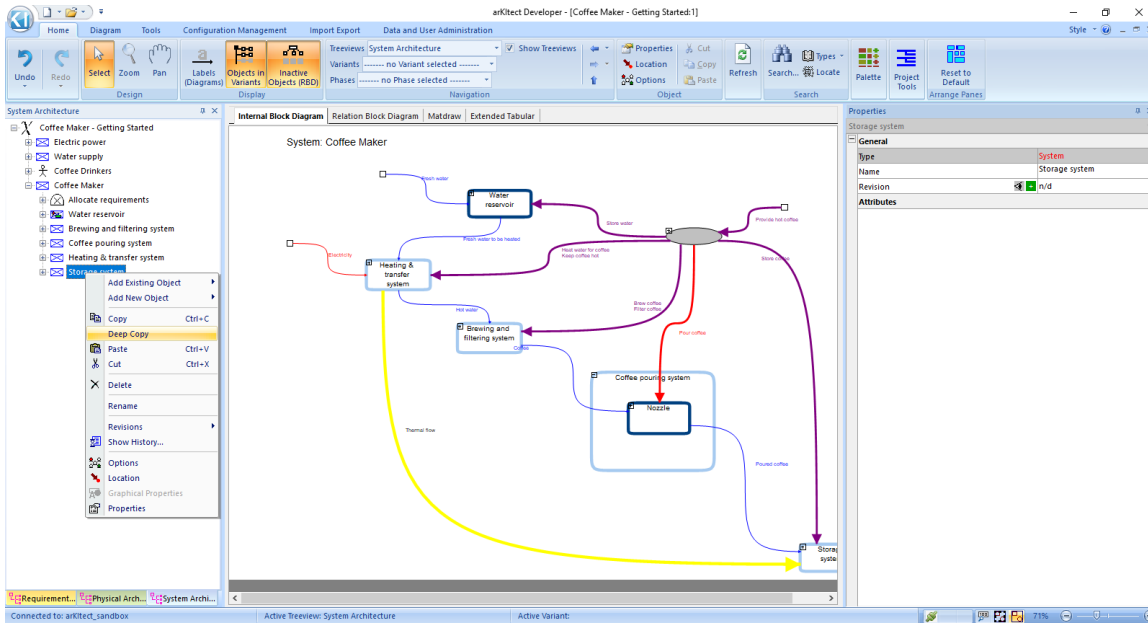


Click on the *Pour coffee* flow to select it; you can see in the **Properties** panel (accessible through Object Panel of the Home Category or using **Ctrl+Shift+P** shortcut) that the flow has an attribute named **Performance**. You can now go and define a value for this attribute by placing the mouse cursor in the empty field. Define the performance value as *Pour minimum 99% of coffee*. Attributes represent additional characteristics or properties related to a given type.



Various object manipulations can also be performed in the **Treeview**. For instance, you can delete and rename objects as well as add references to existing objects and create new objects. There is also a tool called **Deep Copy** that permits you to create an independent duplicate of a chosen

object and of its internal structure. Go to the **Treeview** of the *System Architecture* projection and right-click on the *Storage system* node. Choose **Deep Copy** from the context menu. Now, right-click on the *Coffee Maker* node and choose **Paste**. A duplicate of the *Storage system* object appears on screen. Go inside the duplicate to verify that the internal structure of *Storage system* has been copied: you should find one child component as well as three flows (Thermal, Electricity and Matter flow) consumed in this component.



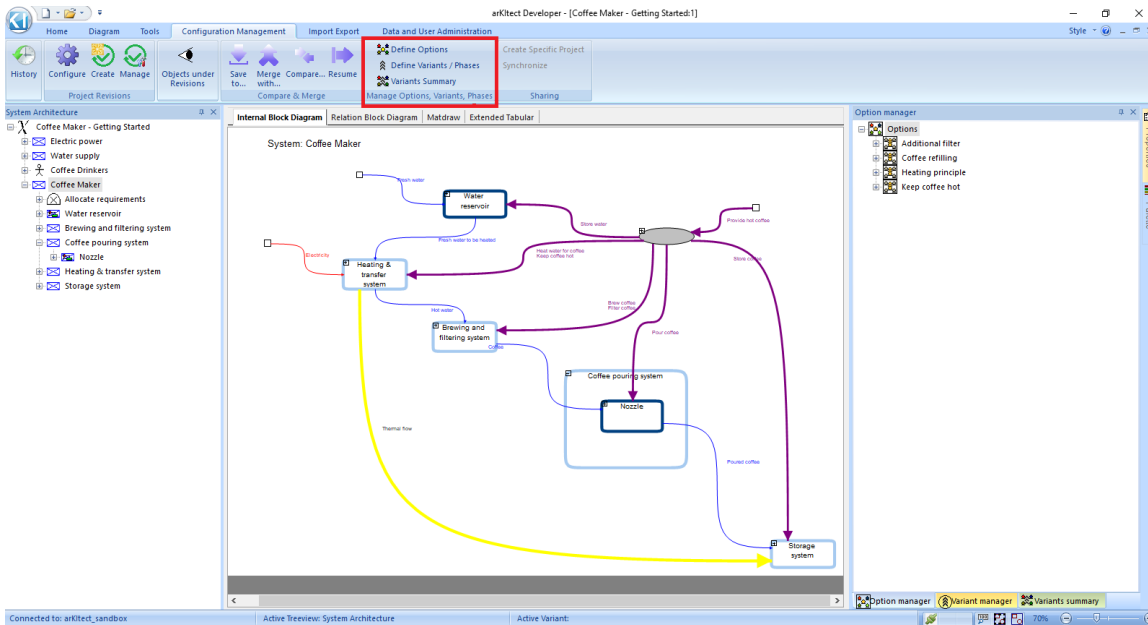
You can now delete this *Storage system* duplicate, it is no longer needed. Note that the deletion of the parent object also removes the child objects.

Next, let us learn about the [Management of Variants](#).

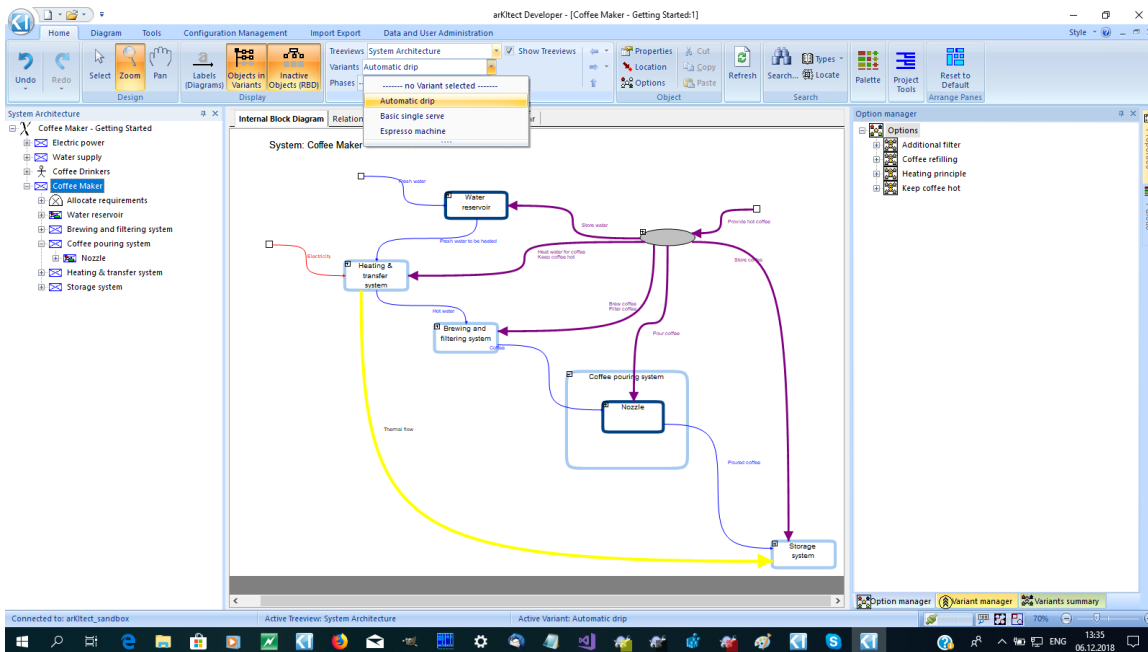
Management of Variants

We shall now explore the means to manage project diversity and variants in **arKitect**. A variant is a subpart of the project; it represents a reduced version of the project or a selection of object instances. Each variant should work independently from the rest of the variants.

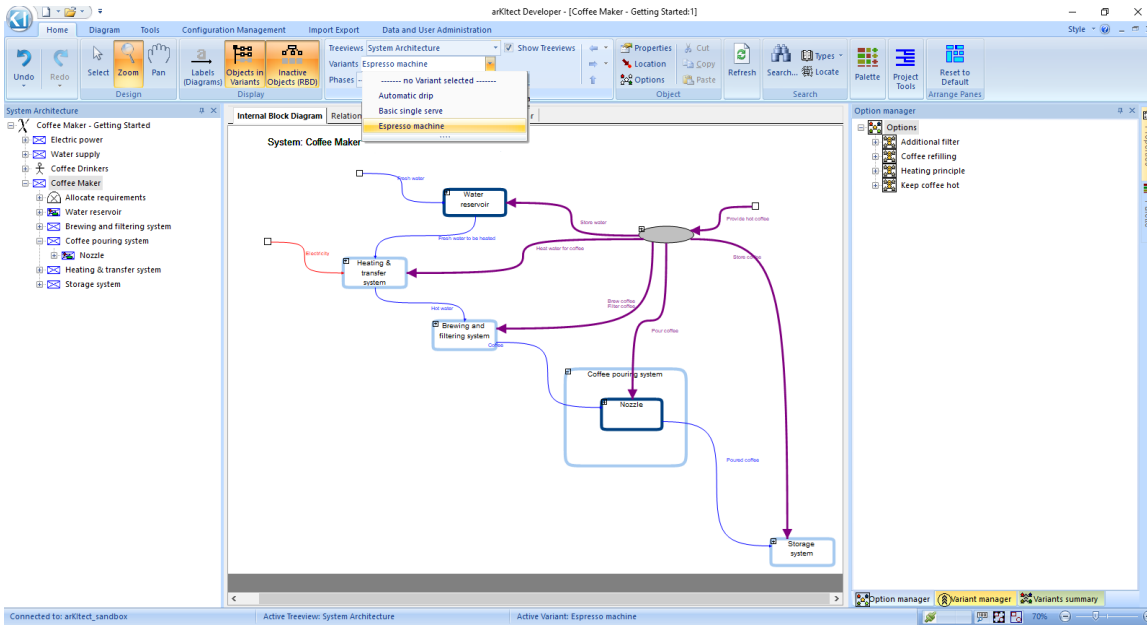
You can close the Palette panel, as it is no longer needed. The main tools needed in this section are the Options Manager, the Variants Manager and the Object Options Window. The **Options Manager** and the **Variants Manager** can be accessed via [Manage Options, Variants, Phases Panel of the Configuration Management Category](#). They are displayed as two tabs in the same panel window. Click on **Options Manager**, **Variants Manager** and **Variants Summary** buttons to display the corresponding panels.



Let us begin by exploring the existing variants. The **Variants** drop-down list in Navigation Panel of the Home Category displays the name of the active variant; you can see that no variant has been selected for the moment. Click on the arrowhead to see the list of variants and select the first variant, *Automatic drip* from the list. There are at the moment no changes in the current diagram view, meaning that all the displayed objects are relevant to the *Automatic drip* variant.



Now, select the *Espresso machine* variant. You can see that the Thermal flow object disappears from view: it is not a part of the *Espresso machine* variant. If you go inside the *Heating & transfer system*, you can see that only the *High pressure heating system* is present.



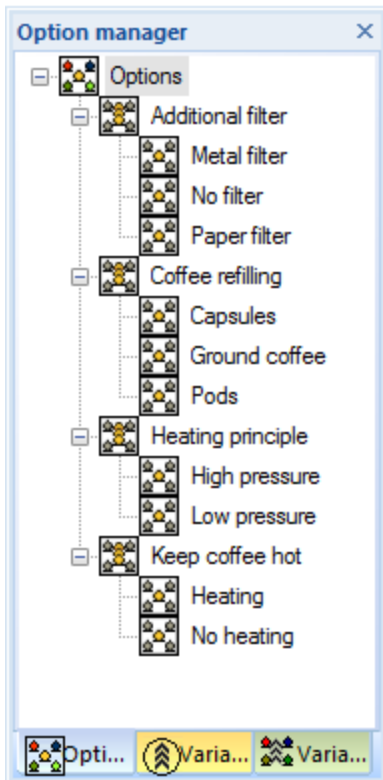
Now go to the last variant, *Basic single serve*. Instead of *High pressure heating system*, *Low pressure heating system* appears in the diagram; however, when you go up one hierarchical level, the *Coffee Machine* diagram view resembles that of the *Espresso machine* (the Thermal flow of *Automatic drip* is not visible).

This demonstration shows that object instances can be filtered from view using the **arKitect** variant management tools. In this way, it is for instance possible to model several alternative technical solutions (such as *High pressure heating system* and *Low pressure heating system*) simultaneously. Afterwards, only the relevant technical solution needs to be displayed for a given variant and the incompatible options can be hidden from view.

The variants management is in practice implemented via two different tools: variants and options. Projects have variants while objects have options. Basically, a variant filters branches of objects in function of the options chosen for object filters. It is thus first necessary to define options before variants can be configured. Afterwards, object instances need to be configured to make the filtering active.

The difference and interlinkages of these components are best explained via a practical example; let us thus define a new variant for the project.

The first step is to go to the Options Manager window.

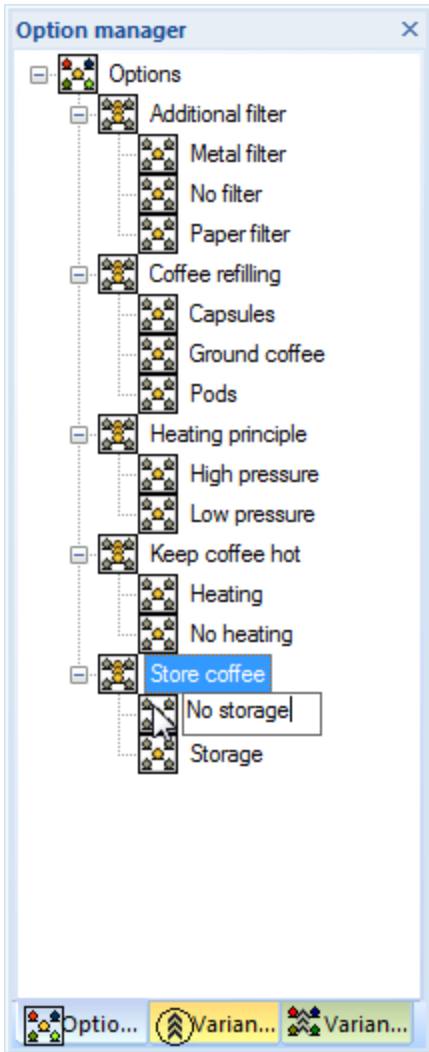


Several option categories have been defined in the project, namely:

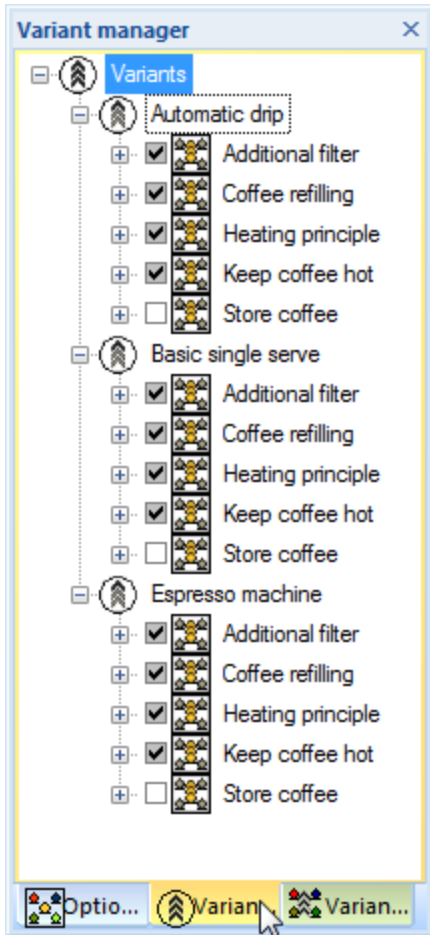
- Additional filter
- Coffee refilling
- Heating principle
- Keep coffee hot

Under these option category nodes you can see the individual options for each category; for instance, the coffee refilling can be done either in the form of *Capsules*, *Ground coffee* or *Pods*. (Click on the category nodes to see the options underneath.)

Let us now add a new option category: right-click on the Options node and choose **Add Option Category**. Name this new category *Store coffee*. Now let us add options to this category. Right-click on *Store coffee* and choose **Add Option**. Add two options and name them *No storage* and *Storage*.

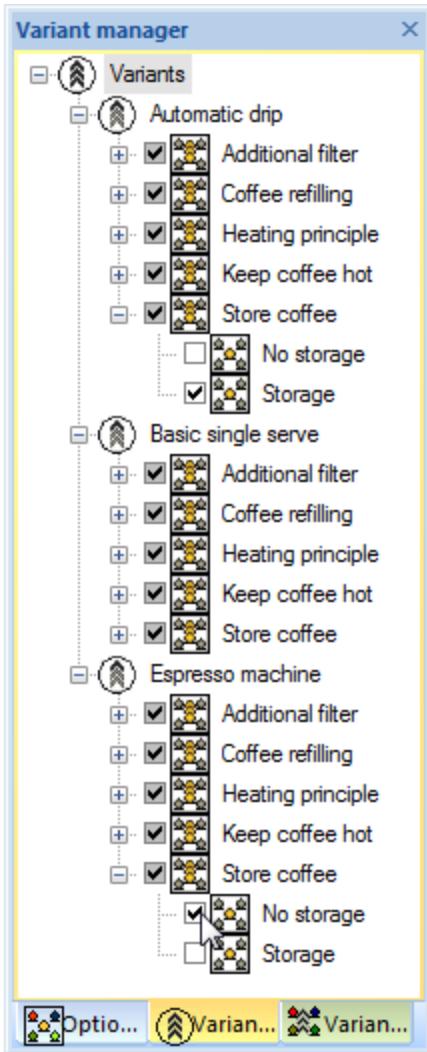


Now that we have defined our options, we can go to the Variants Management tab. (Click on the **Variants Manager** tab at bottom of the panel window.)



We can see the three existing variants listed in the window. In addition, when the variant nodes are opened, the option categories and options linked to each variant are displayed using a system of check-boxes. For instance, the *Automatic drip* is the only variant linked to the *Heating* option.

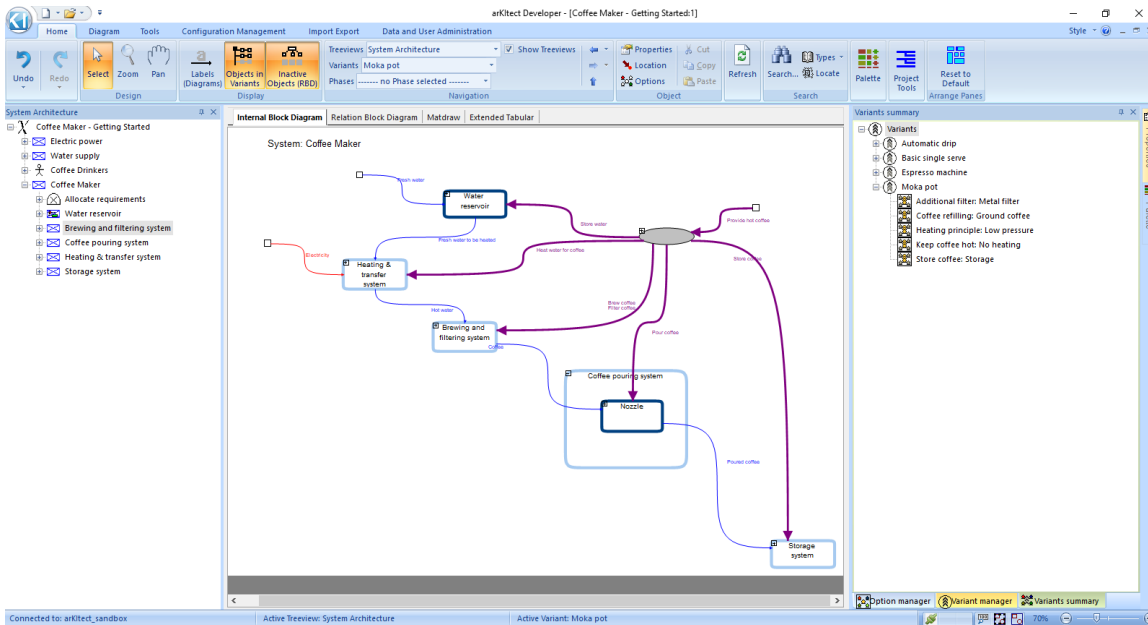
As we only just added the *Store coffee* option category, no variant has yet been linked to this category. Link *Automatic drip* to the *Storage* option and the other two variants to the *No storage* option by checking the corresponding check-boxes.



Let us also define an entirely new variant: right-click on the Variants node and choose **Add Variant** from the context menu. Name this variant *Moka pot*. Check the options linked to this variant, namely:

- Metal filter
- Ground coffee
- Low pressure heating system
- No heating
- Storage

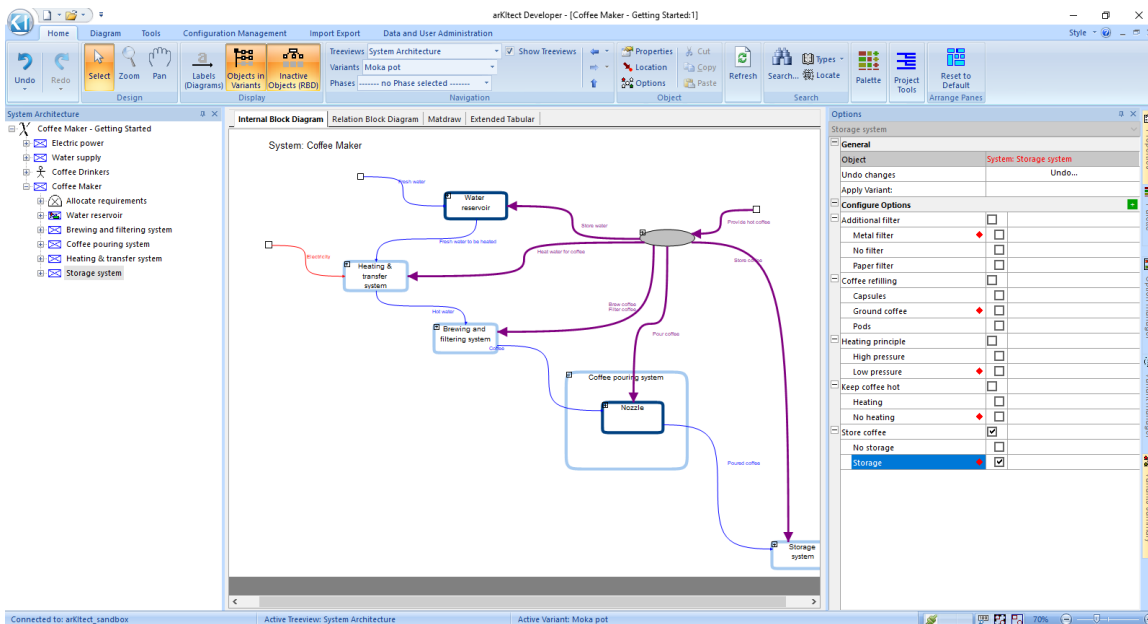
We have now created a new variant and configured it. In order to display the objects related to this variant, go to the **Variants** drop-down list in the [Navigation Panel of the Home Category](#) and choose *Moka pot* from the variants list and set the focus on the *Coffee Maker* object. You can use the **Variants Summary** panel (**Ctrl+Shift+S** shortcut) to visualize at a glance the options linked to a given variant.



Some of the objects of the project have already been configured for previously defined options. For instance, we see no thermal flow from the *Heating & transfer system* to the *Storage system* as this has been linked to the *Heating* option.

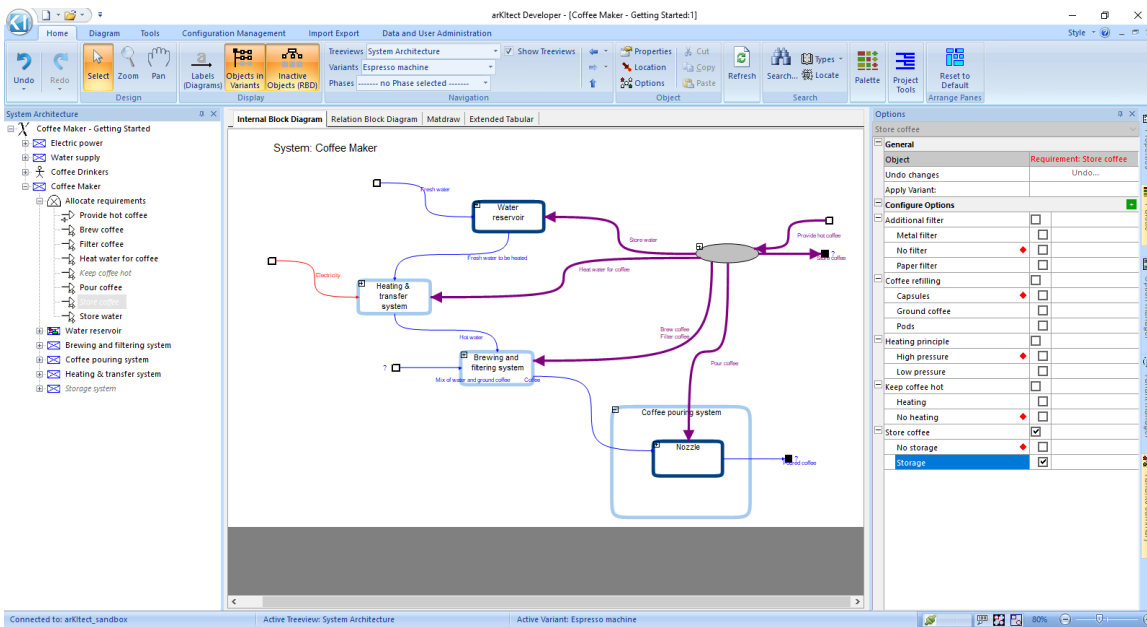
In order to access the Object option configuration use **Object Options** button in the Object Panel of the Home Category (or **Ctrl+Shift+O** shortcut) The Options panel is now displayed at the right-hand side of the user interface.

Click on the *Storage system* object: the Option panel now displays the name of the object and the list of option categories and option. In order to link this object to the *Storage* option, check the corresponding check-box in the Option panel. The other options are not relevant for this object and they do not thus need to be touched.

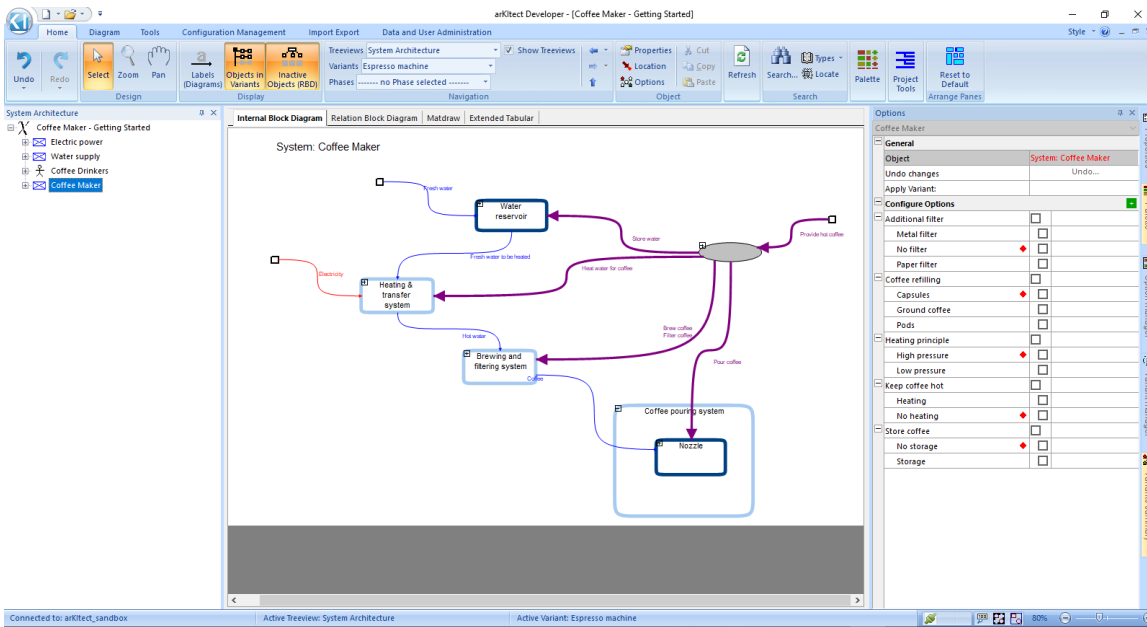


Now, if you change the variant to *Basic single serve* or *Espresso machine* you can see that the *Storage system* is no longer visible for these variants. This is because we created two new options, *Storage* and *No storage*, and both these variants are linked to the *No storage* option. On the other hand, this subsystem is visible for *Automatic drip* as this variant is linked to *Storage*.

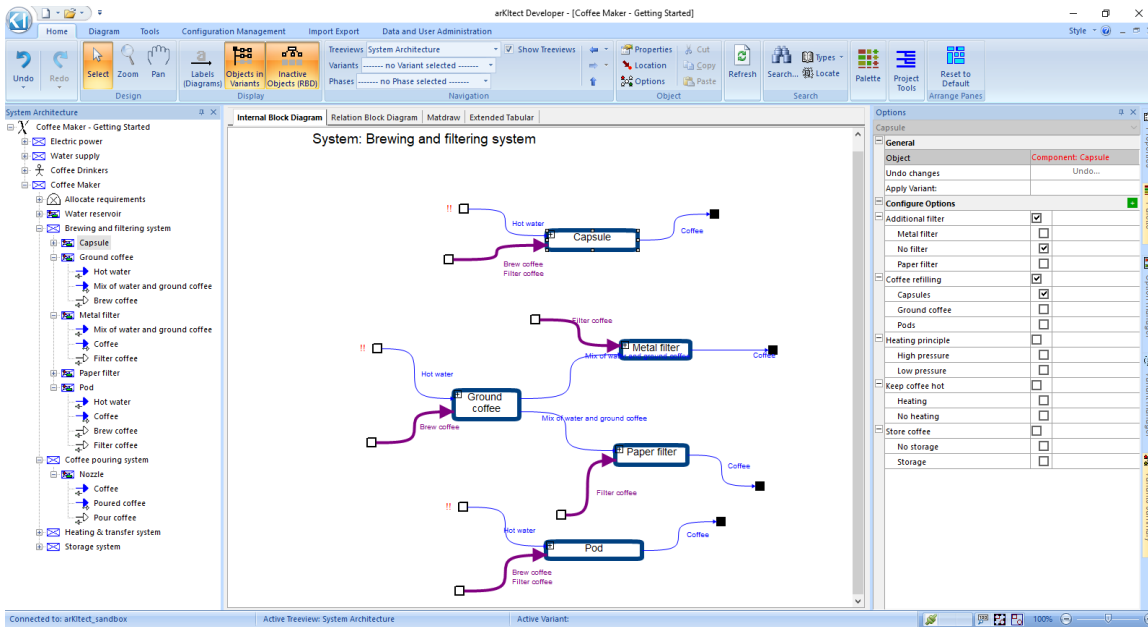
However, there is still a problem in the *Basic single serve* and *Espresso machine* variants: the *Poured coffee* (matter) and *Store coffee* (requirement) flows should also be hidden from view. Go to the Options of the flows and link them to the *Storage* option.



After refreshing the view (F5), these flows should now disappear from view.



You can also explore the configurations of other objects and flows. Return first to the **no Variant selected** state via the Variants drop down list in the Navigation Panel of the Home Category in order to access all the objects of the projection. For instance, the components of the *Brewing and filtering system* are each linked to different options. You can see here that an object can be linked to more than one option at once: for instance, the *Capsule* component is linked to both *No filter* and *Capsule* options.



There are also object instances that are not linked to any option such as *Water reservoir* inside the *Coffee Maker* system. This container is thus displayed for every variant as no variant filtering mechanism is active.

Beyond the Developer Getting Started Guide

There are several **arKitect Developer** functionalities that were not included within the scope of this Getting Started Guide in order to make it easily accessible and to facilitate the first steps in using the software tool. These functionalities are presented in the **arKitect Developer User's Guide**; they include notably

- [Recovering a Project](#)
- [Search and Location Tools](#)
- [Special views, i.e. Relation Block Diagram and Extended Tabular View](#)
- [Collaboration](#)
- [Versioning](#)
- [Advanced Features](#)