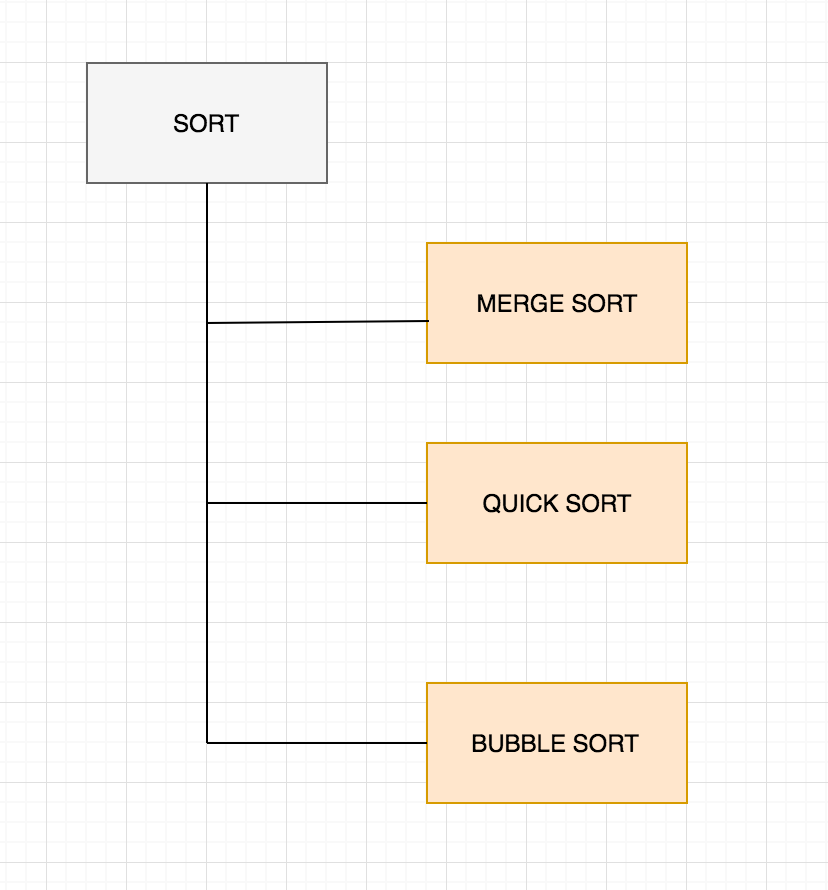
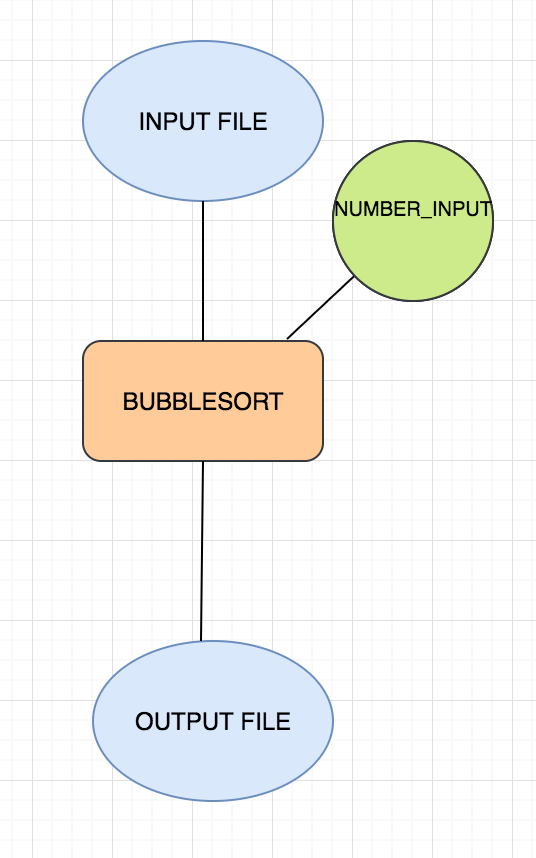
**DOCUMENTATION FOR SCENARIO-2:**

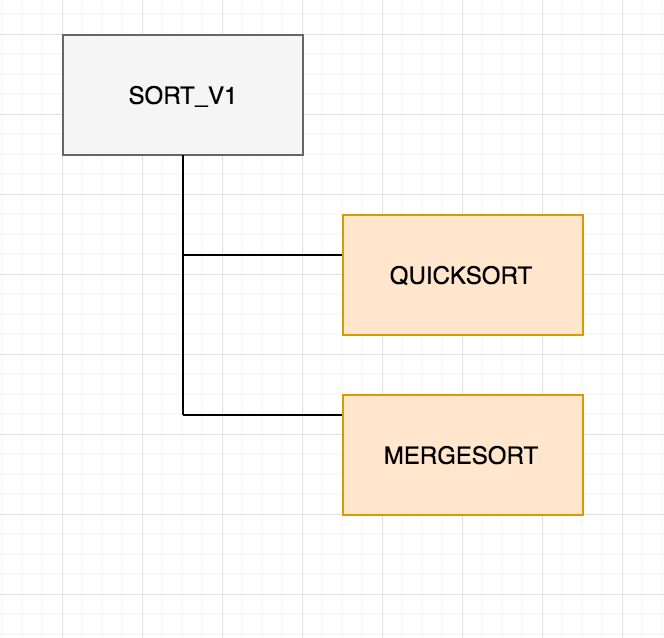
* **What is Scenario-2?**
* **Alice is a user of the WINGS WORKFLOW SYSTEM.**
* **Alice wanted to carry out some sorting of text files.**
* **She created a workflow with a sorting algorithm named Bubble-sort.**
* **She created this new Concrete Component Bubble-Sort and kept it under the Abstract Component SORT.**
* **The SORT Abstract Component has 1 Input, 1 Output.**
* **The new Bubble-Sort concrete Component has 1 Input, 1 Output and 1 parameter.**
* **When the Workflow is published the WINGS Component Catalog exports this scenario as:**

****

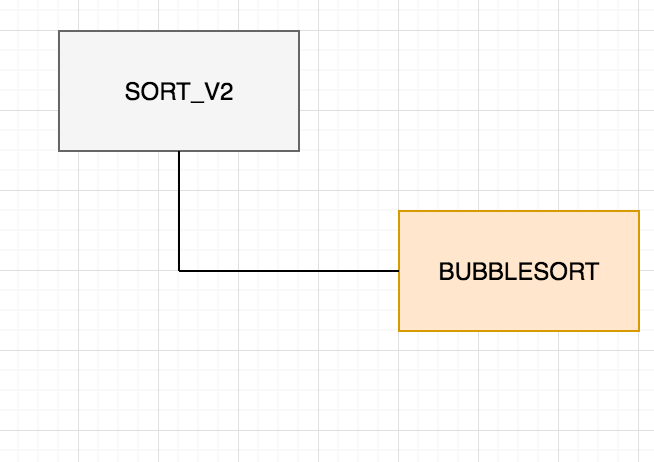
* **Bubble Sort’s concrete component is this:**

****

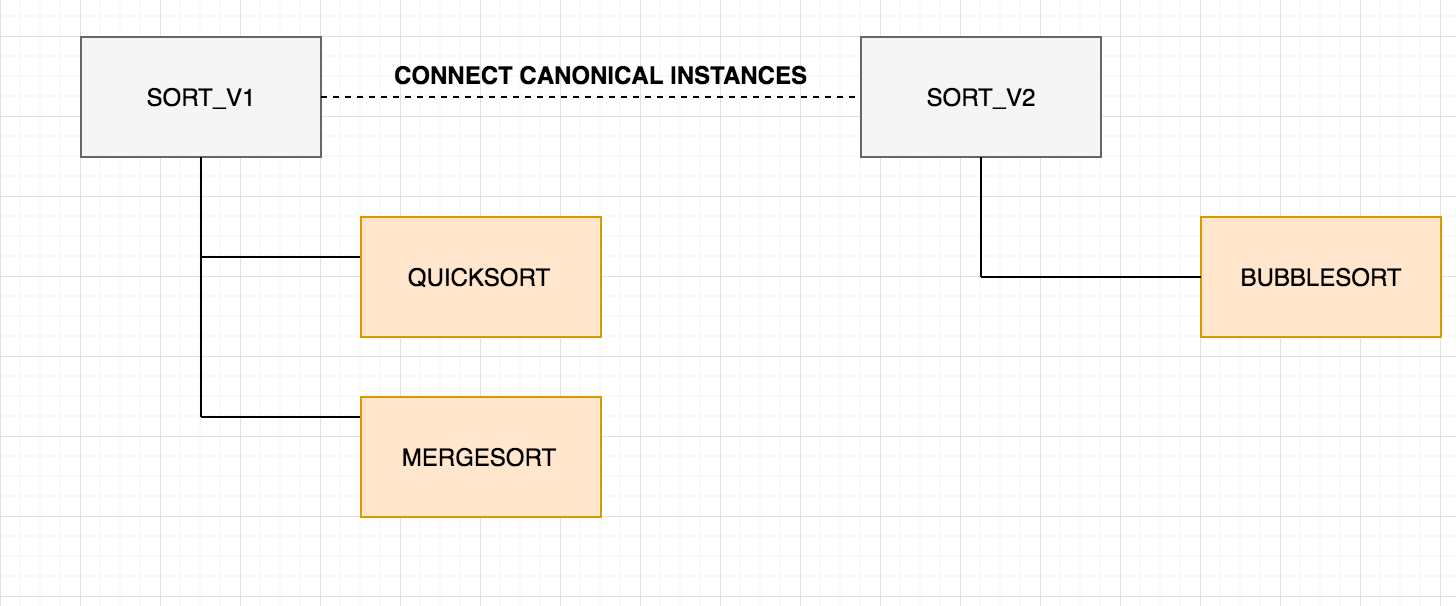
* **The PROBLEM is:**
* **Well, the component catalog currently has Mergesort, QuickSort and the newly added BubbleSort under the same abstract component SORT which is WRONG.**
* **SOLUTION:**
* **We are creating a new TAXONOMY HIERARCHY MODEL (THM) for exporting these changes that will help maintaining a good consistency of components and versioning.**
* **We had export SORT\_V1 and QuickSort and MergeSort under it along with their canonical instances.**

****

* **We will now export SORT\_V2 with BubbleSort under it along with the canonical instance.**

****

* **Further we connect the canonical instances of SORT\_V2 to SORT\_V1 and say that it is a prov:is revision of SORT\_V1.**

****

**VALIDATION TEST EXPECTATIONS:**

* **Scenario-2’s main goal is to handle the versioning of components.**
* **The code’s most important factor is to identify the component coming from the template is an abstract component or a concrete component.**
* **How do you manage this?**

**The way I have done this is. Go through all the components and create 2 hash-sets (one for abstract components and the other for concrete components). The first time I go through all the components, I put only the abstract components in the hash-set1. The second time I go through components, I only add the concrete components in hash-set2 and check if their abstract components are present in hash-set1. If yes, I delete them from there.**

* **Now, we have 2 lists of abstract comps and concrete comps. Hence, we partition the scenario in 2 cases, either abstract or concrete case.**
* **Abstract case:**

1. **First find the abstract names that are common to the one we have. If there are none, we created a new version and export it.**
2. **If we have names that are common and they match the number of inputs and outputs to the ones that are present in the taxonomy model, then no export since there is a match.**
3. **If they do no match, we need to export the latest version to the previous ones and we link a prov: revision Of property also to the latest version from the newest version that we have now.**

* **Concrete case:**

1. **First find the concrete names that are common to the one we have. Next, find the common abstract component names to the one we have with us.**
2. **If both (similar names for the abs and concrete comps) are zero, this is a completely new component, so export the concrete component and its abstract component separately as version1.**
3. **If the similar concrete names are zero but the similar abstract names are not zero, then we have to find if our concrete comp has the same number of inputs and outputs as the abstract comps that are similar. If we find a match anywhere then, we export our concrete component as a subclass of that Abstract component. If one the other hand they don’t match anywhere, then we export the both (abstract and concrete comps) of them with the latest versions of the abstract component.**
4. **If the similar names for the abs comp and concrete comps are non-zero then we check if the concrete comp we have is similar in inputs and outputs to the abs comps in the Taxonomy Model. If there is a match, then we don’t have to export anything. If there is no match, then we have export a new version for the abs comp and concrete comp.**

**Example Cases:**

1. **Let’s say we have a component Bubble-Sort that is a concrete comp of the abs comp SORT. Bubble-Sort has 1 input and 1 output. Now, we export a workflow with this. We will thereby export the Bubble-Sort\_V1 concrete comp and its abs comp SORT as SORT\_V1.**
2. **If we change this component now to have an extra parameter, then we will export Bubble-Sort\_V2 under its SORT\_V2 with 1 input, 1 output and 1 parameter and we further link it to SORT\_V1 and Bubble-Sort\_V1.**
3. **If we now create a new concrete component Quick-Sort. Now, this Quick-Sort is having 1 input and 1 output and its under SORT that has 1 input and 1 output. So Quick-Sort will be exported as a subclass of the SORT\_V1.**
4. **If we have a new abs comp SEARCH with 1 input, 1 output and publish a workflow with this. It will be added as a new component (SEARCH\_V1) in the Taxonomy Model. Now, if we change this abs comp SEARCH and add 2 parameters to it and publish a new workflow. We will have to add it as a new comp (SEARCH\_V2) with linking it to SEARCH\_V1.**