

## Process Model Configurator and Generator Algorithm

We illustrate the Process Model Configurator and Generator algorithm pseudo code in Listing 1. For each system component, our Process Model Configurator and Generator algorithm invokes the BVR API to generate a new resolution model based on the assigned ASIL and it executes the BVR resolution method. We provide the following input parameters to this method: the *VSpec*, *VResolution*, and *VRealization* models (**lines 8-10**) for generating an EPF method plugin that contains the required activities, tasks and guidance to address the ASIL assigned to the component. This is done for each component of the hierarchical platform architecture. If a component contains subcomponents, the same procedure is executed to generate process models for each subcomponent (**lines 12-16**). The algorithm uses a recursive call (**line 14**) for configuring and generating process models for architectural subcomponents. This algorithm was implemented in Java and will be further deployed as an Eclipse plugin.

**Listing 1.** Process model configurator and generator algorithm.

---

```
1 Inputs: ASIL Decomposition (ASILDec), EPF plugin, VSpec, realization;
2 Outputs: a set of process models as EPF method plugins;
3 generate (ASILDec, EPF, VSpec, realization){
4     for (c: ASILDec.components){
5         configure(c, EPF, VSpec, realization);
6     }
7 }
8 configure(c, EPF, VSpec, realization){
9     resolution = BVR.createResolution().setASIL(c.ASIL);
10    BVR.execute(resolution, VSpec, realization);
11    subs []= c.subcomponents();
12    if (subs.size() > 0){
13        for (s: subs){
14            configure(s, EPF, VSpec, realization);
15        }
16    }
17 }
```

---