CyPhy2Desert code logic flow:

Init

initStructuralInterfaceMap(this obsolete now)

traverseDesignContainer

1. create corresponding Desert::Element for the designContainer
2. iterate all the designElement inside this designContaienr

-if the element is Component or ComponentAssembly:

flattenComponent( to get all the property/parameter inside the Component/ComponentAssembly)

-if the element is DesignContainer:

For all properties, processProperties;

For all parameters, processParameters;

For all constants, processConstants;

pre\_processValueFormulas, check whether the CustomFormula/SimpleFormula is the source for SimpleFormula. If it is, then create a Desert variableProperty for transferring the value

-if it is Optional, then add null Element with all assigned VariableProperties, so it will not be excluded with constraints which use the property

addGroup

recursive iterate its children elements

finalize its SimpleFormula and CustomFormula

if it is Alternative or Optional container, processParameters

flatternComponent:

define Com2DesertElement object

call flatternCA()

processProperty

processParameter

processConstant

if the root is ComponentAssembly:

instantiate the element to its type,

set the root=false(it is not directly called from the designContainer)

recursive call flatternCA()

updateInnerVPMap

pre-process SimpleFormula/CustomFormula, if it is the src of another SimpleFormula, then create DesertIface::VariableProperty

process SimpleFormula/CustomFormula and other direct ValueFlows

if it is root(under directly DesignContainer)

process StructuralInterface