CyPhyDesignSpaceRefactorer: Design Space Refactorer Tool for refactoring Design Space Elements.

Overview

This tool plays an important role in allowing the design space refinement process to work. A user should first read and understand the CyPhyDSRefiner tool documentation. During the design space refinement process, a user selects a set of configurations based on the outcome of static and dynamic analysis of all configurations for further refinement. This is done using CyPhy's CyPhyDSRefiner tool, which converts the selected configurations into a newly created refined design space. This space will look like the original design space. The key difference is that any component, component assembly, or design container that is not part of any of the selected design configurations is not included in the refined design space. Secondly, the original static constraints are removed and a new visual constraint is added that directly encodes the configurations selected for design refinement. The reason for this is that the initial configurations were selected only after a detailed constraint satisfaction and dynamics analysis of the original coarse-level design space, and there is no need for repeating this task in the refined design space. For all components that are still part of the refined design space, all connections, ports, and properties are preserved in the generated refined design space. The new refined design space is leaner and a direct representation of the originally selected design configurations. Given its inherent properties, the user can freely refine and expand this design space as different parts are now included or other parts are further elaborated into greater detail.

The CyPhyDesignSpaceRefactorer tool is useful for converting existing components, component assemblies, or design containers into a new design container or component assemblies that can now include new parts in it. In this process, a user first selects a set of components, component assemblies, or design containers and then invokes CyPhyDesignSpaceRefactorer. The tool will create a new design elements depending on what was selected for refactoring. If more than one options are available, then the user is presented with a dialog to make the choice and then appropriate refactoring will be carried out according to user's choice. The good thing about the refactorer tool is that it still preserves all the connections of the refactored design element (or updates them as appropriate).

Below is a summary of what options are applied or choices are presented to the user depending on the context of the design elements chosen for which the refactoring was applied.

Invoke the refactoring interpreter inside **ComponentAssembly**:
CA0: none selected objects:
-No dialog
-New DesignSpace+NewDesignContainer generated under the root folder

CA1: single selected object and it is ComponentAssembly or ComponentRef which refers to a ComponentAssembly:
-Dialog with the following choice:
A: Extracts the elements of the CA
B: Convert it into a new CA
C: Convert it into a new DC(Compound)

CA2: more than one selected objects:
- No dialog
- A new ComponentAssembly generated inside the current ComponentAssembly, and put the all selected objects inside with the connection between them, also additional port generated to connect the object outside the new CA

Invoke the refactoring interpreter inside **DesignContainer**:
DC0: none selected objects:
- usage info message shown up.
- no action

DC1. single selected object:
- DC1_CA: if the object is ComponentAssembly or ComponentRef which refers to a ComponentAssembly:
  - as same as CA1

-DC1_DC: if the object is designContainer:
-DC1_DC_C: the type is compound
-DC1_DC_C_0: there is no DesignContainer inside
-Dialog with the following choice:
A: Extracts the elements of the DC
B: Convert it into a new DC(Alternative)
C: Convert it into a new CA
-DC1_DC_C_1:there is DesignContainer inside
-Dialog with the following choice:
A: Extracts the elements of the DC
B: Convert it into a new DC(Alternative)

-DC1_DC_A: the type is Alternative
- No dialog
- Generate a new Alternative designContainer and put the selected DC inside it

-DC1_DC_O: the type is Optional
-DC1_DC_O_1: there is only one element inside
- Dialog with the following choices:
A: Convert the optional element into mandatory(extract out)
B: Convert it into a new DC(Alternative)
- DC1_DC_O_2: there are more than one element inside
  - same as DC1_DC_A

-DC1_other(Component or ComponentRef which refers to Component):
- Dialog with the following choice:
A: Convert it into a new DC(Alternative)
B: Convert it into a new CA

DC2: more than one selected objects:
-DC2_1: only DC objects selected,
- no dialog
- convert the objects into a new generated designContainer of Alternative type

-DC2_2: if there is any CA, Component and ComponentRef selected:
-Dialog with the following choice:
A: Convert it into a new DC(Alternative)
B: Convert it into a new CA
-Result:
A: new DC generated and all selected objects: DC, CA, Component and ComponentRef inside;
B: only convert CA, Component and ComponentRef into new generated CA, for any designContainer, there is no action.