

Core to SAL Additional Info

Node Linking

Since ID attribute is not allowed by sal's DTD, the nodes are linked by the xpointer syntax.
For instance (relaxing relation to assign statement) :

```
<Change xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="extended">
  <Source xlink:type="locator"
    xlink:href="relation.cdl_xml.xml#Relation 24"
    xlink:label="Expr 41DB8"
    xlink:role="Relation term"
    xlink:title="Relation term" />
  <Target xlink:type="locator"
    xlink:href="buffer_modular.cdl_flat_xml#xpointer(//*[@2]//*[@2]//*[@2]//*[@2]//*[@0]
1//*[@0]//*[@1]//*[@1]//*[@1])"
    xlink:label="SIMPLEDEFINITION 14"
    xlink:role="assignment"
    xlink:title=" assignment " />
  <Connection xlink:type="arc"
    xlink:from=" Expr_41DB8"
    xlink:to="SIMPLEDEFINITION14"
    xlink:show="other"
    xlink:title="relation term relaxing"
    xlink:actuate="onRequest" />
</Change>
```

The connection types

1 Relation Relaxing

1.1 **Relation to Enable**

When a relation term is moved to the guard (enable) section (no next-state variable)

| | |
|-----------------|---|
| Source | The relation term |
| | xlink:title = "Relation term" ; xlink:role="Relation term" |
| Target | The relevant guard expression |
| | xlink:title="Guard expression" ; xlink:role="Guard expression" |
| Connection Type | "relation term relaxing" |
| | xlink:show="other"; xlink:title = "relation term relaxing" ; xlink:actuate="onRequest" |

1.2 **Relation to Assign**

When a relation term is moved to the assign section (simple boolean term)

| | |
|-----------------|---|
| Source | The relation term |
| | xlink:title = "Relation term" ; xlink:role="Relation term" |
| Target | The relevant assign expression |
| | xlink:title="Assignment" ; xlink:role="Assignment" |
| Connection Type | "relation term relaxing" |
| | xlink:show="other"; xlink:title = "relation term relaxing" ; xlink:actuate="onRequest" |

1.3 **Transition split caused by relation**

The relation could not be relaxed.

| | |
|-----------------|--|
| Source | The relation expression |
| | xlink:title = "Relation" ; xlink:role="Relation" |
| Target | The new hidden transition |
| | xlink:title = "Hidden transition" ; xlink:role="Transition" |
| Connection Type | "relation split" |
| | xlink:show=""; xlink:title = "relation split" ; xlink:actuate="onRequest" |

For each next state variable reference in the relation, where the variable has no assignment in the transition a non-deterministic assignment is added

| | |
|-----------------|--|
| Source | Next-state variable references in the relation |
| | Xlink:title = "variable refernce" ; xlink:role="variable references" |
| Target | The non-deterministic assignment |
| | Xlink:title = "non deterministic assignment" ; xlink:role="assignment" |
| Connection Type | "relation split next-state" |
| | xlink:show=""; xlink:title = "relation split next-state"; xlink:actuate="onRequest"EW |

2 Partial synchronic handling

2.1 **By flattening**

The additional information from the flattener is used. Target nodes ID's are replaced from the core flattened nodes to sal nodes.

2.2 **By module splitting**

The participate modules are split

| | |
|-----------------|---|
| Source | The partial combination |
| | xlink:title = "partial ModCombin" ; xlink:role="ModCombin" |
| Source | The original module |
| | xlink:title = "variable refernce" ; xlink:role="variable references" |
| Target | The new module |
| | xlink:title = "Module" ; xlink:role=" Module" |
| Connection Type | "partial sync" |
| | xlink:show="other"; xlink:title = " partial sync"; xlink:actuate="onRequest" |

The local parameters are moved to the combination scope

| | |
|-----------------|---|
| Source | The partial combination |
| | xlink:title = "partial ModCombin" ; xlink:role="ModCombin" |
| Source | The original variable declaration |
| | xlink:title = "Variable" ; xlink:role="Variable" |
| Target | Variable definition in the split modules (global and not local) |
| | xlink:title = "Variable" ; xlink:role="Variable" |
| Connection Type | "partial sync" |
| | xlink:show="other"; xlink:title = " partial sync"; xlink:actuate="onRequest" |

| | |
|-----------------|---|
| Source | The partial combination |
| | xlink:title = "partial ModCombin" ; xlink:role="ModCombin" |
| Source | The original variable declaration |
| | xlink:title = "Variable" ; xlink:role="Variable" |
| Target | Variable definition in the combination module |
| | xlink:title = "Variable" ; xlink:role="Variable" |
| Connection Type | "partial sync" |
| | xlink:show="other"; xlink:title = " partial sync"; xlink:actuate="onRequest" |

The combination module is changed from partial to sync and async

| | |
|-----------------|---|
| Source | The original combination |
| | xlink:title = "partial ModCombin" ; xlink:role="ModCombin" |
| Target | The new combinations |
| | xlink:title = "ModCombin" ; xlink:role="ModCombin" |
| Connection Type | "partial sync" |
| | xlink:show="other"; xlink:title = " partial sync"; xlink:actuate="onRequest" |

3 Transition renaming

A new module is created with the transition name changed

| | |
|-----------------|--|
| Source | The rename definition xlink:title = "Rename" ; xlink:role="Rename" |
| Target | The new module xlink:title = "Module" ; xlink:role="Module" |
| Connection Type | "transition renaming" xlink:show="other"; xlink:title="transition renaming"; xlink:actuate="onRequest" |

4 No Hold previous

4.1 **Non Array variables**

Non deterministic assignments are added

| | |
|-----------------|--|
| Target | The non-deterministic assignment xlink:title = "Assignment" ; xlink:role="Assignment" |
| Connection Type | "no hold previous" xlink:show="other"; xlink:title="no hold previous"; xlink:actuate="onRequest" |

5 Remove Multiple ModCombin Modules

5.1 **change modcombin**

replace inst modcombin that calls another module with the modcombin of that module

| | |
|-----------------|--|
| Source | The original Modcombin xlink:title = "Modcombin" ; xlink:role="Modcombin" |
| Source | The original module xlink:title = "Module" ; xlink:role="Module" |
| Target | The new modcombin node xlink:title = "Modcombin" ; xlink:role="Modcombin" |
| Connection Type | "no multiple modecombin modules" xlink:show="other"; xlink:title="no multiple modecombin modules"; xlink:actuate="onRequest" |

The local parameters are moved to the new combination scope

| | |
|-----------------|--|
| Source | The original module xlink:title = "Module" ; xlink:role="Module" |
| Target | Variable definition in the combination module xlink:title = "Variable" ; xlink:role="Variable" |
| Connection Type | "no multiple modecombin modules" xlink:show="other"; xlink:title="no multiple modecombin modules"; xlink:actuate="onRequest" |

6 General

6.1 **Cojoin Flag**

Cojoin is copied to the enable section of each transition.

| | |
|-----------------|---|
| Source | The cojoin definition |
| | xlink:title = "Cojoin flag" ; xlink:role = "Cojoin flag" |
| Target | The relevant expression in the guard (enable) section |
| | xlink:title = "Guard expression" ; xlink:role = "Guard expression" |
| Connection Type | "cojoin" |
| | xlink:show = "other"; xlink:title = "cojoin"; xlink:actuate = "onRequest" |

6.2 **Define**

Each use of define macro is expanded

| | |
|-----------------|---|
| Source | The define definition |
| | xlink:title = "Define" ; xlink:role = "Define" |
| Target | The expanded node |
| | xlink:title = "Expr" ; xlink:role = "Expr" |
| Connection Type | "define" |
| | xlink:show = "other"; xlink:title = "define"; xlink:actuate = "onRequest" |

6.3 **Const**

Each use of define macro is expanded

| | |
|-----------------|--|
| Source | The const definition |
| | xlink:title = "Const" ; xlink:role = "Const" |
| Target | The expanded node |
| | xlink:title = "Expr" ; xlink:role = "Expr" |
| Connection Type | "const" |
| | xlink:show = "other"; xlink:title = "Const"; xlink:actuate = "onRequest" |