GroMEt Base Schema $\mathcal{G}$

Wire → Port → Box

element *container*

Containment
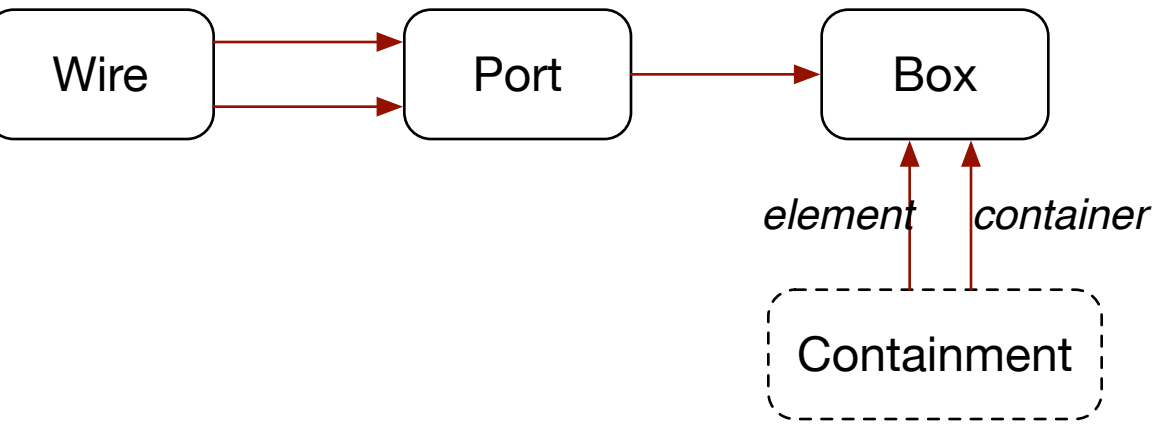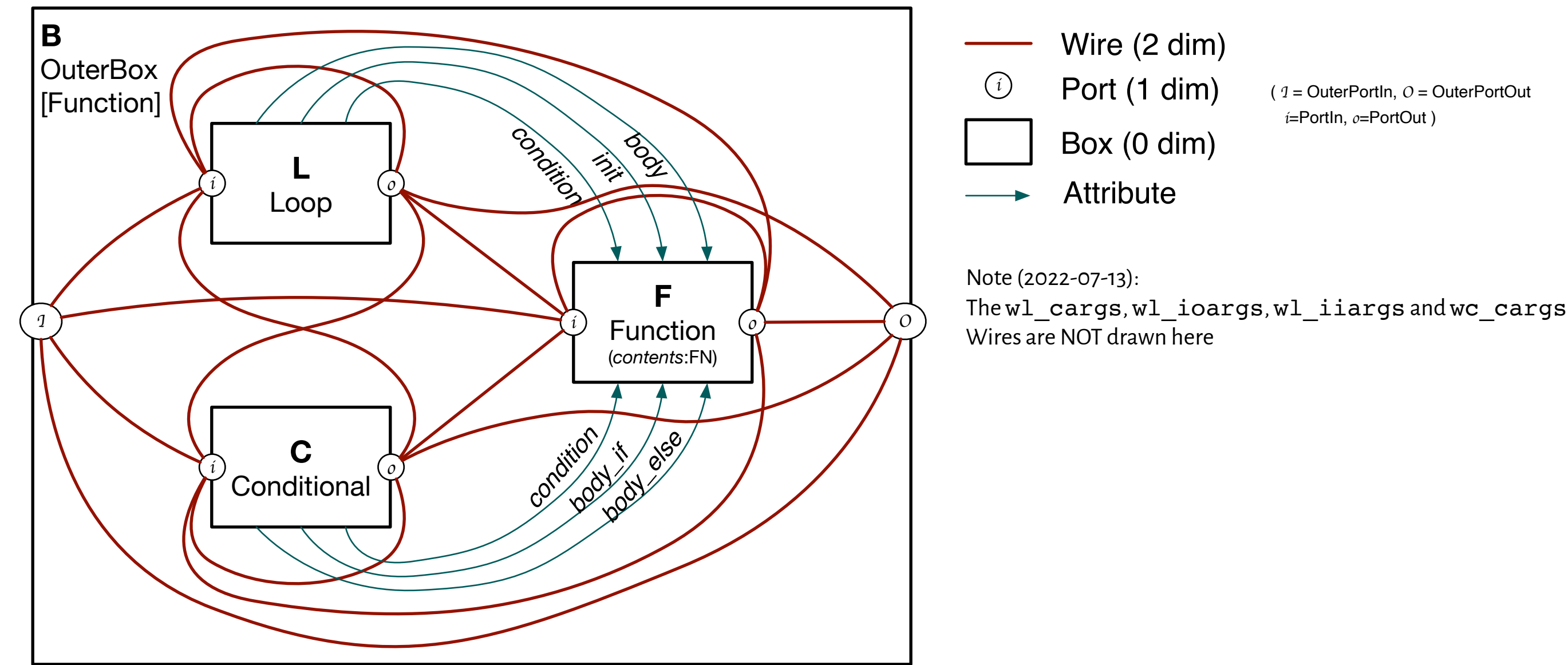
GroMEt FunctionNetwork $\mathcal{FN}$ v0.1.1

Instance of Base $\mathcal{FN} \in Set^{\mathcal{G}}$

(Instance of Base Schema as a category-theoretic Wiring Diagram)

B
OuterBox
[Function]

L Loop

F Function
(contents:FN)

C Conditional

condition *body* *init*
condition *body_if* *body_else*

— Wire (2 dim)
ⓘ Port (1 dim) ($\mathcal{I}$ = OuterPortIn, $\mathcal{O}$ = OuterPortOut, i=PortIn, o=PortOut)
☐ Box (0 dim)
→ Attribute

Note (2022-07-13):
The `wl_cargs`, `wl_ioargs`, `wl_iiargs` and `wc_cargs`
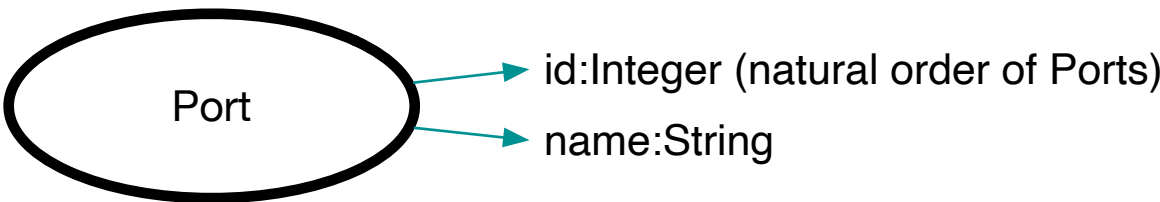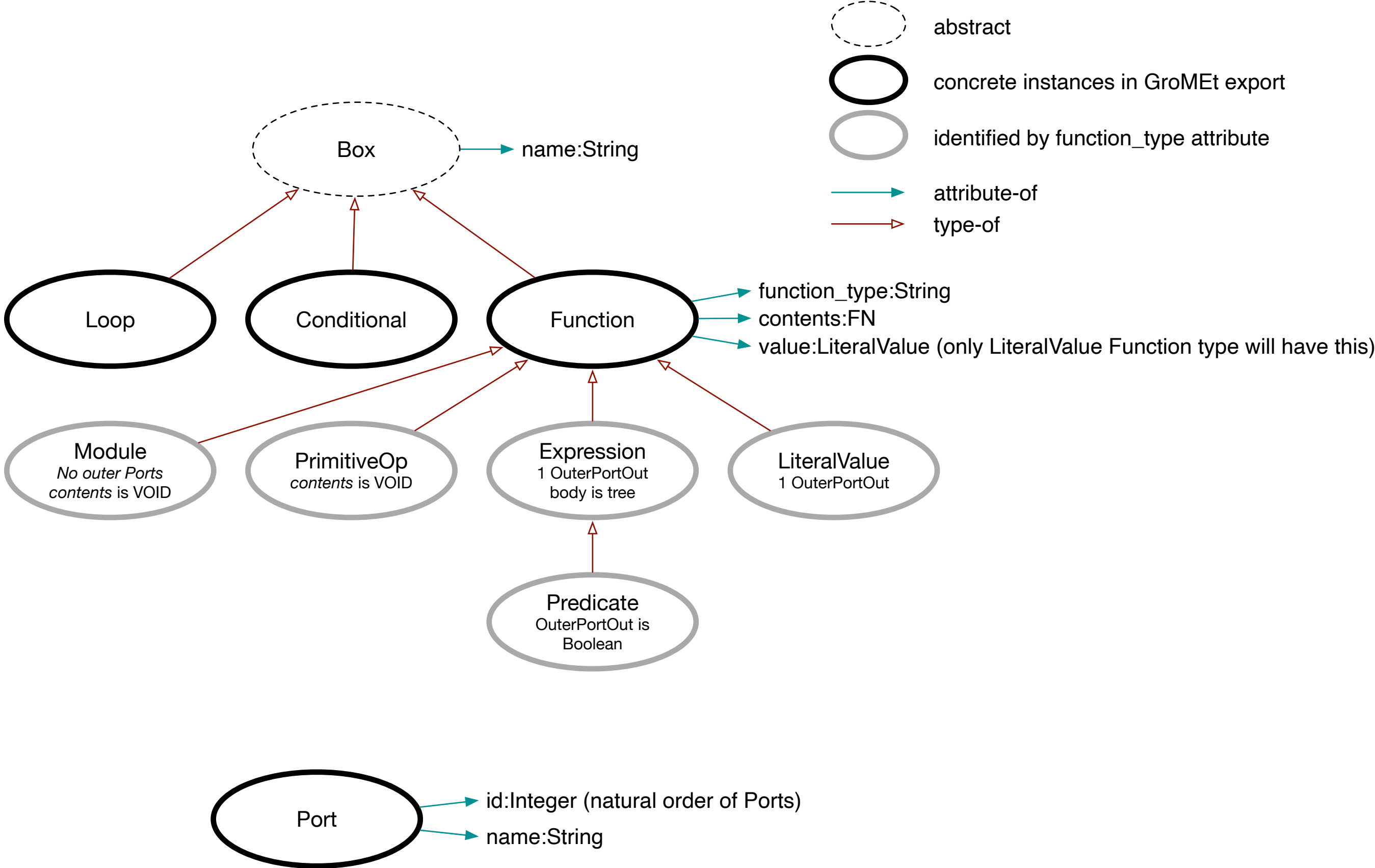Wires are NOT drawn here

$\int_{\mathcal{G}} \mathcal{FN}$ Derived schema for $Set^{\mathcal{G}}/\mathcal{FN}$

(Can be directly interpreted as an ER (Entity-Relation) database schema)

Box Types

Functions have their own hierarchy that will not be explicitly represented in the GroMEt export,
but instead represented by a *function_type* (String) attribute associated with the Function Box.
The most general type is Function, which includes no constraints on number of input and output Ports.
When a box has FN contents, the Box type will be the type of the outer box of the FN.
These structural constraints are not enforced by the schema but would need to be validated
These Box types will also have different attributes, as represented by the green arrows pointing at the attribute names:types.
The smaller text within the ovals describe some additional constraints imposed by a given type.
All child Box types inherit attributes of parents (even if hey don't use/fill them) as well as structural/contents constraints.
Module is special and only appears that the "top" level FN.

- - - abstract
— concrete instances in GroMEt export
◯ identified by function_type attribute
→ attribute-of
⇢ type-of

Box → name:String

Loop   Conditional   Function → function_type:String
→ contents:FN
→ value:LiteralValue (only LiteralValue Function type will have this)

Module
No outer Ports
contents is VOID

PrimitiveOp
contents is VOID

Expression
1 OuterPortOut
body is tree

LiteralValue
1 OuterPortOut

Predicate
OuterPortOut is
Boolean

Port → id:Integer (natural order of Ports)
→ name:String

Note (2022-07-13): The red color indicates changes that need to be made in the SKEMA code base to match the (corrected) association of src and tgt; once fixed in the code base, these can be made black.

wlopi *src*
tgt
wll *src*
tgt
wlf *tgt*
src
wlc *tgt*
src
wlopo
src
tgt

pil *box* bl

wl_cargs
(src loop condition
Predicate gets values
from tgt loop input Ports)

wl_iiargs
(src loop init function
input Ports get values
from tgt loop input Ports)

wl_ioargs
(src loop input Ports get
values from tgt init
function outputs)

pol *box*
tgt
src
tgt
src
src
tgt

condition [Predicate]
init [Function]
body [Function]

opi
b *box*
wopio *tgt*
src
opo *box*

wfopi *tgt* *src*
wfl *src* *tgt*
wff *src* *tgt*
wfc *tgt*
wfopo *tgt*
src

pif *box* bf *contents*→ FN
pof *box*

bijection

wcopi *tgt* *src*
wcl *src*
wcf *tgt* *src*
wcc *src* *tgt*
wcopo *src* *tgt*

pic *box* bc
poc *box*

wl_cargs
src *(src conditional condition*
Predicate gets values
from tgt loop input Ports)
tgt

condition [Predicate] *body_if* [Function] *body_else* [Function]

Notes:

Naming Conventions: All type names start with their schema syntactic type:
op = OuterPort
opi = OuterPort Input Ports, opo = OuterPort Output Ports
p = Port
the next Port name letters:
i = in, o = out ; followed by l = Loop, f = Function, c = Conditional
w = Wire
For 3-lettered Wire names, the last two letters are:
src -> tgt, with l = Loop, f = Function, c = Conditional
B = Box
followed by l = Loop, f = Function, c = Conditional

Special Wire cases:
(1) **wopio**: Wire from OuterPortIn to OuterPortOut (i.e., passthrough)
(2) **wl_cargs** and **wc_cargs** represent a map (as Wires) from the arguments of the condition (**cargs**) Predicate (src) of the **L**oop or **C**onditional to the PortIn's (tgt) of the Loop or Conditional Box.

Similarly, **wl_iiargs** and **wl_ioargs** represent a map from the input and output (respectively Ports of the loop init Function to the loop input Ports.

The background color patches group Ports, Wires and Boxes that are present depending on whether Function (green), Loop (blue) or Conditional (orange) is present.