Simple Modeling Language - Cheatsheet

Box:

Syntax:

```
<Boxname1>: <Boxtype1> @<Tag>{
  <Boxcontent>
}
```

DSL:

```
Box1:entity @pos(5,5) {
    Box2: entity {
        iAmAString: string
    }
    Box3: entity {}
}
```

Result:

Types*: **entity**, **enumeration**, device, actor, program, network, cluster, container, thread, module, component, function, artifact, tier, area, **layer**, **slice**, **state**, transition, compound, vgroup, hgroup, type

Link:

Syntax:

```
<Boxname1>: <Boxtype1> {
  <AssociationName>:
    <Boxname2><SimpleCardinality>
}

<Boxname2>: <Boxtype2> {
  <AssociationName>:
    <Boxname2>[<Cardinality>]
}

<Boxname3>: <Boxtype3> {
  <AssociationName>: <Boxname2>
    @dock("<Port1>","<Port2>"
}
```

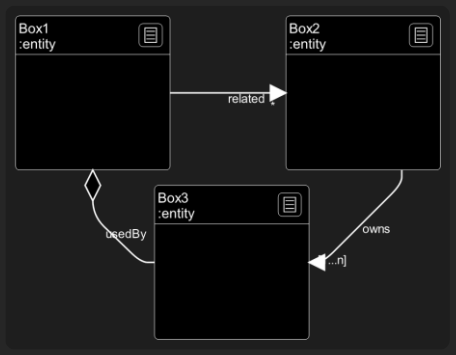
DSL:

```
Box1:entity @pos(5,5){
    related: Box2*
}

Box2: entity @pos(40,5){
    owns: Box3 [1...n]
        @dock("br","rm")
}

Box3: entity @pos(23,27){
    usedBy: Box1
        @compose("lm","bm")
}
```

Result:



Types*: **dock**, **inherit**, **compose**, call, call-back, one-way, generic

Tag:

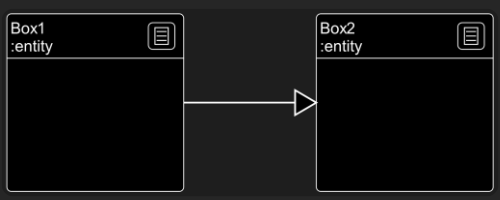
Syntax:

```
<Boxname1>: <Boxtype1> @<Tag> {
  <Attribute>: <AttributeType> @<Tag>
}
```

DSL:

```
Box1:entity @pos(5,5) {
    Box2 @inherit
}
Box2:entity @pos(40,5){}
```

Result:



Types*: Ordered, unique, **pos, dock**, **inherit**, **compose**, call, call-back, one-way, generic, hint, start, end

* Bold types are currently supported