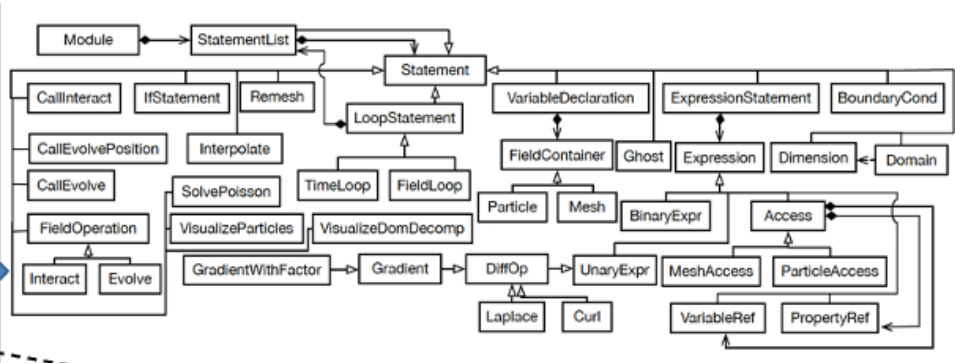


```

rhs -> vortex_stretching_m =
(vorticity_mesh -> vorticity_m.∇)
velocity_mesh -> velocity_m +
nu * Δ vorticity_mesh -> vorticity_m

```

## OpenPME



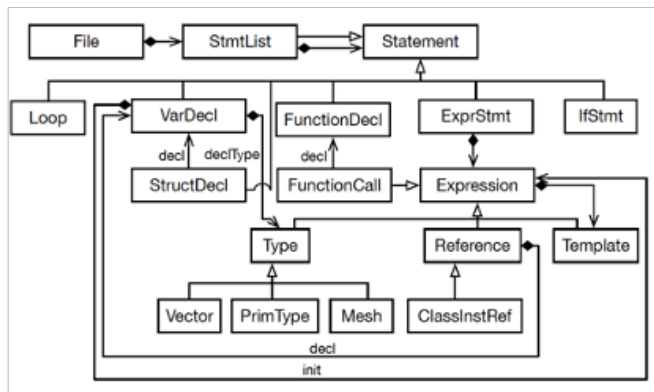
IR

```

for mesh node decl <no type> loopNodeM in rhs
loopNodeM -> vortex_stretching_m [ 0 ] = ...
loopNodeM -> vortex_stretching_m [ 1 ] = ...
loopNodeM -> vortex_stretching_m [ 2 ] = ...

```

## Metamodel for particle-mesh simulations



## Metamodel for OpenFPM C++

```

while (mloop_iterator_h5a0.isNext())
{
    g_dwp.template get<rhs>(key)[x]=
    fac1*(g_vort.template get<vorticity>(key.move(x,1))[x]+
    g_vort.template get<vorticity>(key.move(x,-1))[x])+
    fac2*(g_vort.template get<vorticity>(key.move(y,1))[x]+
    g_vort.template get<vorticity>(key.move(y,-1))[x])+
    fac3*(g_vort.template get<vorticity>(key.move(z,1))[x]+
    g_vort.template get<vorticity>(key.move(z,-1))[x]) -
    2.0f*(fac1+fac2+fac3)*
    g_vort.template get<vorticity>(key)[x]+
    fac4*g_vort.template get<vorticity>(key)[x]*
    (g_vel.template get<velocity>(key.move(x,1))[x]-
    g_vel.template get<velocity>(key.move(x,-1))[x])+
    fac5*g_vort.template get<vorticity>(key)[y]*
    (g_vel.template get<velocity>(key.move(y,1))[x]-
    g_vel.template get<velocity>(key.move(y,-1))[x])+
    fac6*g_vort.template get<vorticity>(key)[z]*
    (g_vel.template get<velocity>(key.move(z,1))[x]-
    g_vel.template get<velocity>(key.move(z,-1))[x]);

    g_dwp.template get<rhs>(key)[y]=
    g_dwp.template get<rhs>(key)[z]=
}

```

## OpenFPM