```
package apd;


platform aircraft_preliminary_design{


      component HeatSource {
            var heat Real;
      }

      component ElectricLoad {
            var v Real ;
            var i Real ;
      }

      component FuelTank {
            view Mechanical {
                  var fret Real ;
                  var fout Real ;
                  var tret Real ;
                  var tout Real ;
                  assumption {
                        fret <= fout ;
                  }
                  guarantee {

                  }
            }
      }

      component HeatLoad {
            view Mechanical {
                  var fin Real ;
                  var fout Real ;
                  assumption {

                  }
                  guarantee {
                        fin = fout ;
                  }
            }
            view Thermal {
                  var cf Real ;
                  var fin Real ;
                  var tin Real ;
                  var fout Real ;
                  var tout Real ;
                  var heat Real;
                  assumption {
                  }guarantee {
                        fin*tin*cf + heat = fout*tout*cf ;

                  }


            }
      }
```

```
component Engine extends HeatSource {
      view Thermal {
            var hnom Real ;
            var tmin Real ;
            var tmax Real ;
            var tin Real ;

            assumption{

            }
            guarantee {
                  if ( (tin <= tmax) and (tin >= tmin))
                        (heat = hnom)
                  else
                        (heat = 0) ;
            }
      }
      view Mechanical {
            var fnom Real ;
            var tmin Real ;
            var tmax Real ;
            var fin Real ;
            var tin Real;
            assumption {

            }
            guarantee {
                  if ( tin <= tmax and tin >= tmin )
                        (fin = fnom)
                  else
                        (fin = 0) ;
            }
      }
}

component Splitter {
      view Mechanical {
            //var fmax  Real ;
            var fin Real ;
            var f1 Real ;
            var f2 Real ;
            assumption {
                  //(f1 + f2 <= fmax) ;
            }
            guarantee {
                  (fin = f1 + f2) ;
            }
      }
      view Thremal {
            var tin Real ;
            var t1 Real ;
            var t2 Real ;
            assumption{

            }
            guarantee{
                  (t1 = tin) and (t2 = tin) ;
            }
      }
}




component Generator extends HeatSource{

      view Electrical {
            var vnom Real ;
            var pmax Real ;
            var i Real ;
            var v Real ;
            assumption {
                  (i*vnom <= pmax) ;

            }
            guarantee{
                  (v = vnom) ;
            }
      }

      view Thermal {
            var vnom Real ;
            var eff Real ;
            var i Real ;



            assumption {

            }

            guarantee {
                  (heat = vnom*i*(1-eff)) ;
            }

      }

}




component Load extends HeatSource, ElectricLoad {
      view Electrical {
            var vnom Real ;
            var pnom Real ;

            assumption {

            }
            guarantee {
                  if ( (v>=9/10*vnom) and (v <= 11/10*vnom) )
```

```
                      (vnom*i = pnom)
                  else
                        (i = 0) ;
            }
      }


      view Thermal {
            var vnom Real ;
            var pnom Real ;
            var eff Real ;
            assumption {

            }
            guarantee {
                  if ( (v>=9/10*vnom) and (v <= 11/10*vnom)  )
                      (heat = pnom*eff)
                  else
                        (heat = 0) ;
            }
      }
}

component ElectricPump extends ElectricLoad {
      var vnom Real ;
            var power Real ;
      view Electrical {

            assumption {

            }
            guarantee {
                  if ( (v>=9/10*vnom) and (v <= 11/10*vnom)  )
                      (vnom*i = power)
                  else
                        (i = 0) ;
            }
      }

      view Thermal {
            var eff Real ;
            var f Real ;
            var cf Real ;
            var tin Real ;
            var tout Real ;
            var power Real ;
            assumption {

            }
            guarantee {
                  if ( (v>=9/10*vnom) and (v <= 11/10*vnom)  )
                        (f*tin*cf + power*(1-eff) = f*tout*cf)
                  else
                        tout = tin ;
            }
      }
```

```
            view Mechanical {
                  var pdrop Real ;
                  var density Real ;
                  var eff Real ;
                  var f Real ;
                  var power Real ;
                  var fin Real ;
                  var fout Real ;
                  var vnom Real ;
                  assumption {

                  }
                  guarantee {
                              if ( (v>=9/10*vnom) and (v <= 11/10*vnom)  )
                           (power*eff*density = pdrop*f) and (fout = f) and (fin = f)
                              else
                              (fin = 0) and (fout = 0) and (power = 0) ;
                  }
            }
      }



      component HeatExchanger {
            view Mechanical {
                  var fin Real ;
                  var fout Real ;

                  assumption {

                  }
                  guarantee {
                        (fout = fin) ;
                  }
            }

            view Thermal {
                  var eff Real ;
                  var tair Real ;
                  var tin Real ;
                  var tout Real ;
                  assumption {

                  }
                  guarantee {
                        (tout = tin - eff*(tin - tair)) ;
                  }
            }
      }



      //Rules
      assertion totalHeat {
            forall h HeatLoad  {  (  h.heat = Sum{ c HeatSource | connected(c,h)  ,
c.heat }  )  } ;
      }
      assertion current {
            forall g Generator {
                  g.i = Sum{ l ElectricLoad | connected(g,l) , l.i }
            };
      }

      assertion positiveheat {
            forall h HeatSource {
                  h.heat >= 0
            } ;
      }

      validity fuelTankTemperature {
            forall ft FuelTank {
                   (ft.fret > 0) -> (ft.tret <= ft.tout + 5) and (ft.tret >= ft.tout
- 5)
            } ;
      }


}

architecture arch from aircraft_preliminary_design {
      fuelTank FuelTank ;
      pump ElectricPump ;
      heatLoad HeatLoad ;
      splitter Splitter ;
      engine Engine ;
      hex HeatExchanger ;
      g Generator ;
      l Load ;
      hs HeatSource ;


      fuelTank.tout = 15 ;
      pump.vnom = 270 ;
      pump.pdrop = 6900000 ;
      pump.f = 20/10;
   pump.density = 800;
      pump.eff = 6/10;
      pump.cf = 2300;
      heatLoad.cf = 2300 ;
   engine.tmax = 112 ;
      engine.tmin = 51 ;
      engine.fnom = 7/10 ;
   engine.hnom = 40000;
   hex.tair = 0 -30 ;
      hex.eff = 55/100 ;
      g.vnom = 270 ;
      g.pmax = 270000  ;
      g.eff = 85/100 ;
      l.vnom = 270;
      l.pnom = 150000  ;
      l.eff = 85/100 ;
      hs.heat  = 60000 ;
      connected(fuelTank.tout,pump.tin) ;
      connected(fuelTank.fout, pump.fin) ;
      connected(pump.tout, heatLoad.tin) ;
      connected(pump.fout, heatLoad.fin) ;
      connected(heatLoad.tout, splitter.tin) ;
      connected(heatLoad.fout, splitter.fin) ;
      connected(splitter.f1, engine.fin) ;
      connected(splitter.t1, engine.tin) ;
      connected(splitter.f2, hex.fin) ;
      connected(splitter.t2, hex.tin) ;
      connected(hex.tout,fuelTank.tret);
      connected(hex.fout,fuelTank.fret) ;
      connected(g.v,pump.v);
      connected(g.v, l.v) ;
      connected(l,heatLoad);
      connected(g,heatLoad);
      connected(hs,heatLoad);
      connected(engine,heatLoad);

}
```