Controller component

```
thread ADSController
features
Efriction: in data port; Eslope: in data port;
Ex_env: in data port; Ev_env: in data port;
Ea_env: in data port; Ecurr_xego: in data port;
Ecurr_vego: in data port;
Econtl_aego: out data port;... ...
properties
   Dispatch_Protocol => Periodic;
   EnvAADL::isController => true;
end ADSController;

thread implementation ADSController.impl
subcomponents
... ...
action:data {Data_Model::Real_Range=>0..0;};
properties
   EnvAADL::Action=> (-1,1);
annex behavior_specification
{** variables
control_a_ego:Base_Types::Float;... ...
states
ST1:initial complete state; ST2: state; ... ...
transitions
ST1-[on dispatch]-> ST2;
ST2-[]-> ST1{
   d:=(Ex_env-Ex_ego)
   if(d-50> 0)
      control_a_ego:= 3-10*0.01*Efriction
      *cos(Eslope); action:=1    end if;
   if(d-50 <=0)
      control_a_ego:= -3-10*0.01*Efriction
      *cos(Eslope); action:=-1    end if;
   Econtl_aego:=control_a_ego};... ...
**}; end ADSController.impl;
```

Environment component

```
system Environment
features
Efriction:out data port; Eslope:out data port;
Ex_env:out data port; Ev_env:out data port;
Ea_env:out data port;... ...

properties
EnvAADL::isEnvironment => true;
end Environment;

system implementation Environment.impl
subcomponents
friction: data Dynamics_Types::dynamic;
            {Data_Model::Real_Range=>0..0;};
slope: data Dynamics_Types::dynamic;
            {Data_Model::Real_Range=>0..0;};
x_env: data Dynamics_Types::dynamic;
            {Data_Model::Real_Range=>50..50;};
v_env: data Dynamics_Types::dynamic;
            {Data_Model::Real_Range=>5..5;};
a_env: data Dynamics_Types::dynamic;
            {Data_Model::Real_Range=>0..0;};
... ...
connections
C1:port friction->Efriction
C2:port slope->Eslope
C3:port x_env->Ex_env
C4:port v_env->Ev_env
C5:port a_env->Ea_env
... ...
properties
EnvAADL::ContinuousDynamics =>
  "x_env'=v_env; v_env'=a_env...";
end Environment.impl;
```