**Design Decisions:**

* Each Actor can only process on message at a time.
* Simultaneous continuous behavior in on actor is allowed.
  + Equations on same variable are summed for continuous behavior.
* There is only on type of message and this message can contain continuous and computational behavior at the same time.

**Semantics:**

DVar : set of all discrete variables names.

CVar : set of all continuous variables names.

Mtd : set of all method declarations.

DelayStatus : true: actor is delayed. False: actor is not in delayed

Each method is defined as the tuple .

**Statements**

+

**Auxiliary functions**

inv: in which inv(x,m) returns the invariant of continuous method of m of the actor x.

guard: in which guard(x,m) returns the guard of continuous method of m of the actor x.

ode: in which ode(x,m) returns the ordinary differential equations of continuous method of m of the actor x.

action:in which action(x,m) returns the transition method of continuous method of m of the actor x.

D : in which D returns the delay variable for actor ID.

**Operational Semantics**

The global state is a function and DS is the discrete state and is defined as and HS is the continuous state and is defined as

**Transitions**

For simplicity discrete variables are omitted.

(CAssignment)

(Continuous behavior)

(Delay)

(continuous behavior deprecation)

(processing resume)

In other rules is changed to and DelayStatus must be false.

**Hybrid Translation**

From ) and CVar To

**Simplifications and assumptions**

No Parameter for methods.

Only one guard and invariant in continuous behavior.

Issues:

* What happens when to actor have message in their queue?
* Do we still have coarse grain semantic for atomic execution of instantaneous statements?