

Semantics of Asbru_{light}



Utrecht, 14.02.2002, Michael Balser, Christoph Duelli

Principles

- semantics of operational part of Asbru
- notation in statecharts

Structure of Semantics

1. hierarchy of plans
2. plan state model
3. time annotations for plan activation
4. plan body
 - do-seq/unordered/parallel/any-order
 - do-all/some
5. upward/downward propagation
6. interface to data abstraction unit

Missing Features

Basic Constructs

- intentions/effects

Conditions

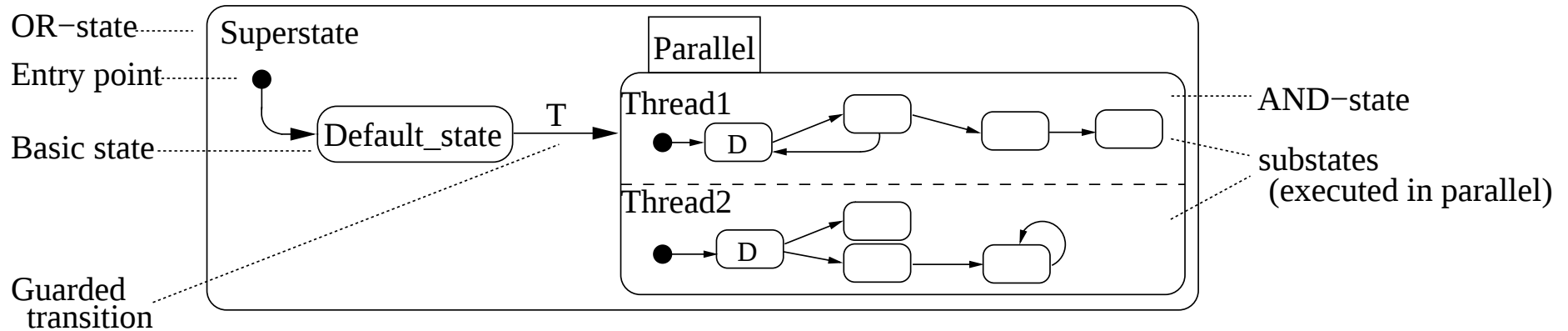
- setup
- suspend/resume
- new style of activation

Data Handling

- context (data abstraction)
- local variables

Control Options

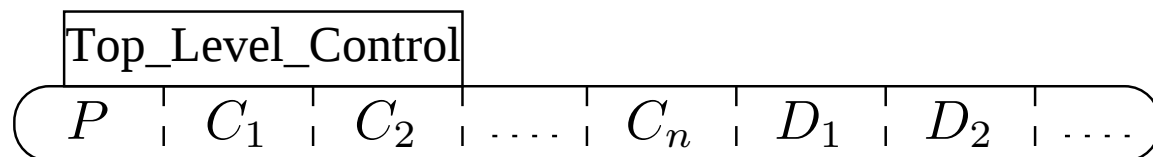
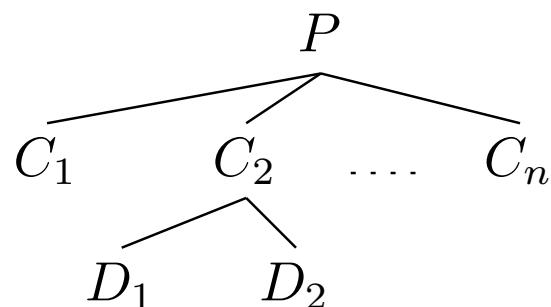
- do-cyclical
- do-one
- retry-aborted
- on-abort
- wait-for-optional



- transitions `events[conditions]/actions`
- condition `in(<State>)`
- micro/macro steps

Important Decision

Parent and sub plan *instances* are executed in parallel

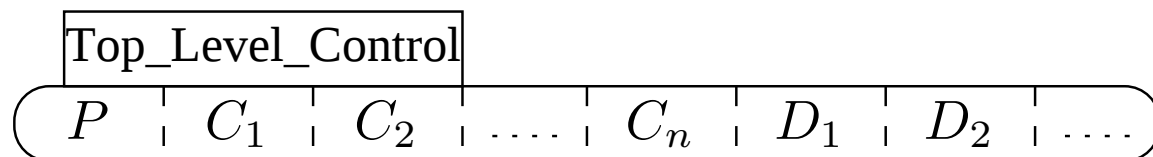
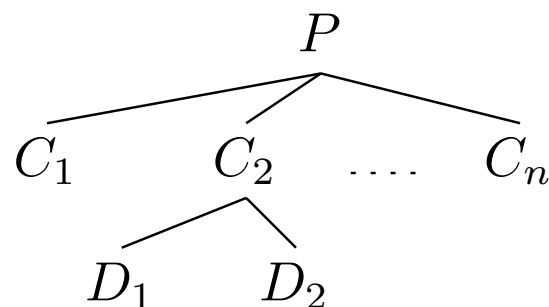


⇒ hierarchy is flattened

⇒ downward propagation is possible

Important Decision

Parent and sub plan *instances* are executed in parallel

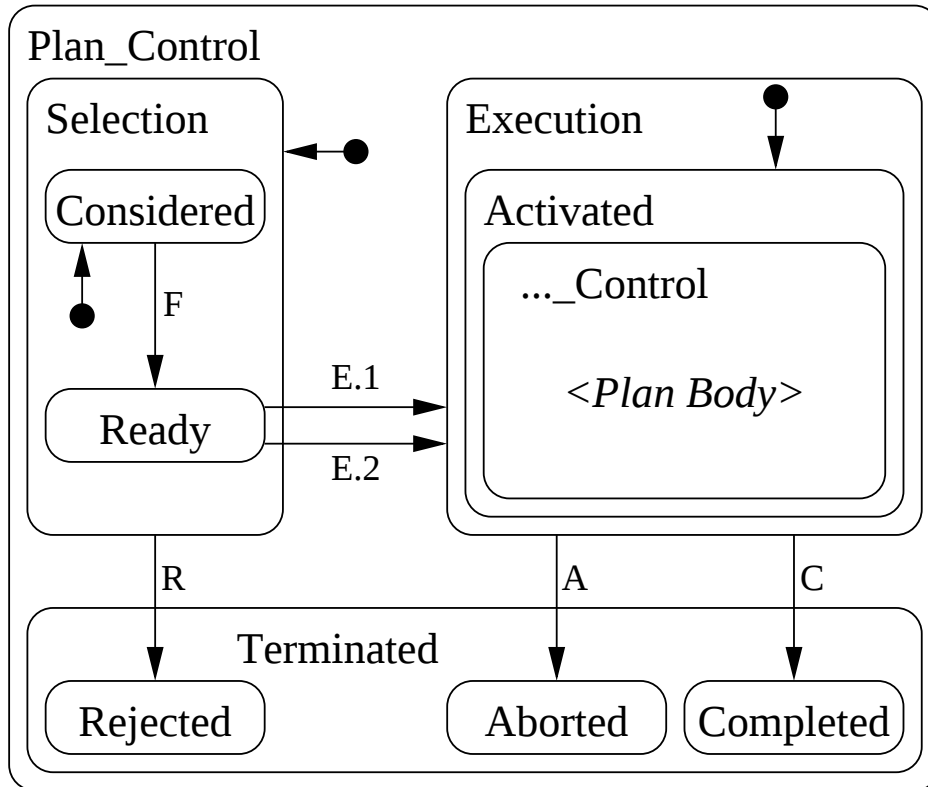


⇒ hierarchy is flattened

⇒ downward propagation is possible

Decision not without problems (see later)

Overall Plan Control



$F : [filter_cnd = true]$

$R : [filter_cnd = false]$

$E.1: [act_cnd = automatic]$

$E.2: select[act_cnd = manual]$

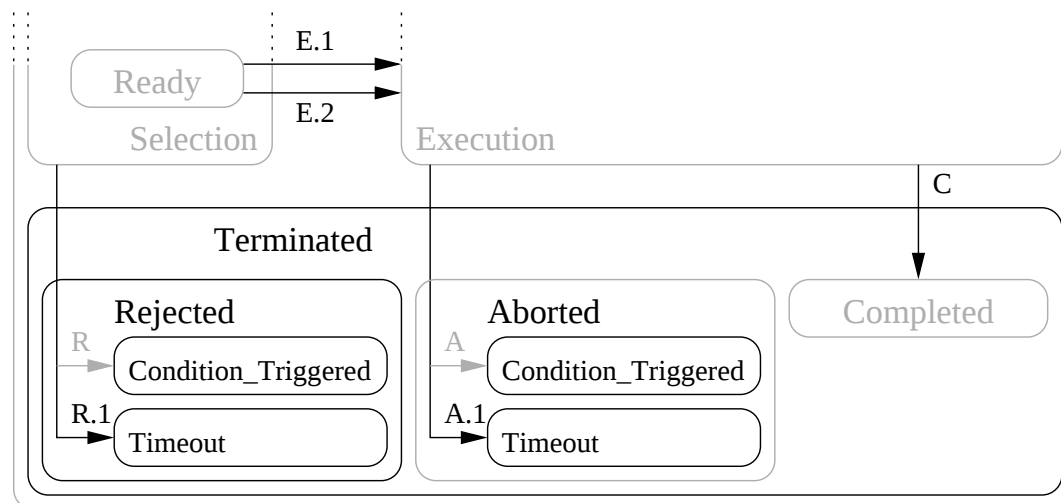
$A : [abort_cnd = true]$

$C : [complete_cnd = true]$

- controlling statechart for sub plans in Activated state

Time Annotations for Activation

$[[\text{ESS}, \text{LSS}], [\text{EFS}, \text{LFS}], [\text{MinDu}, \text{MaxDu}], \text{REF}]$



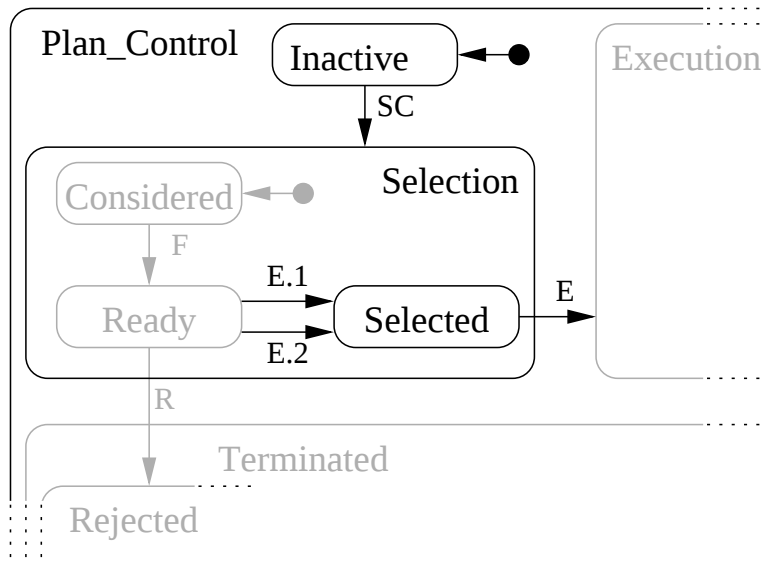
$R.1 : [\text{LSS} + \text{REF} < \textit{time}]$

$C : [\textit{complete_cnd} = \text{true} \quad]$
 $\wedge \text{EFS} + \text{REF} \leq \textit{time} \leq \text{LFS} + \text{REF}$
 $\wedge \text{MinDu} \leq \textit{time} - \textit{start_time} \leq \text{MaxDu}$

- conditions with time annotations in data abstraction unit

Synchronization

Sub Plan

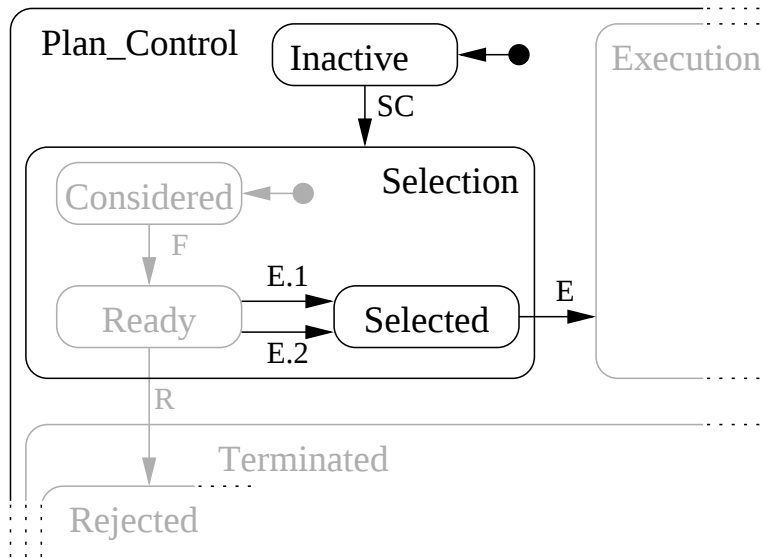


SC: consider

E : activate[...] / ...

Synchronization

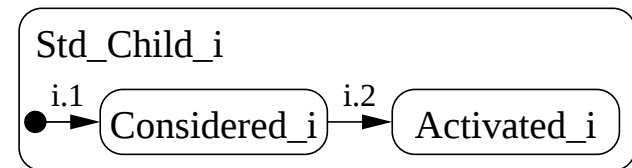
Sub Plan



SC : consider

E : activate[...]/...

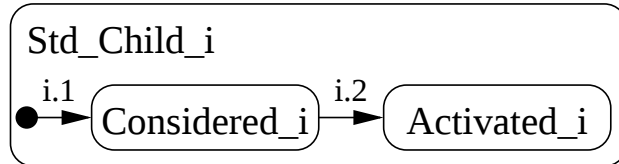
Standard Parent Control



$i.1$: / C_i .consider

$i.2$: [**in**(C_i .Selected)] / C_i .activate

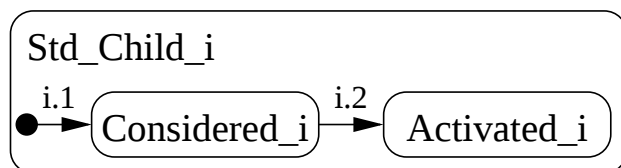
Control Operators



$i.1: / C_i.consider$

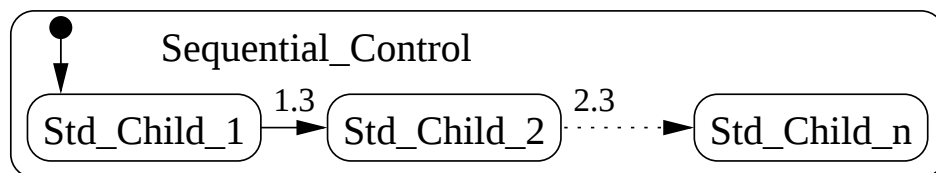
$i.2: [\mathbf{in}(C_i.Selected)] / C_i.activate$

Control Operators



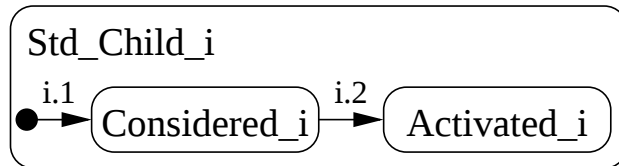
i.1: $/ C_i.consider$

i.2: $[in(C_i.Selected)] / C_i.activate$



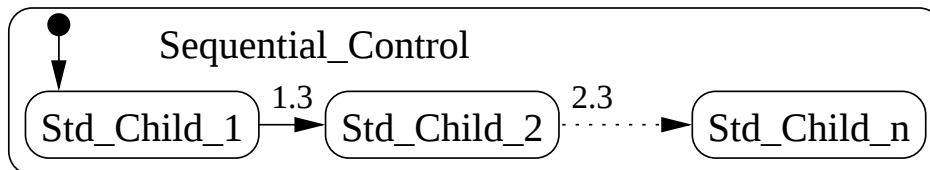
i.3: $[in(C_i.Terminated)]$

Control Operators

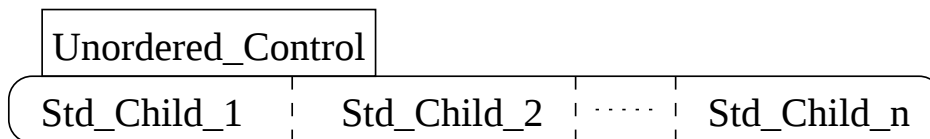


$i.1: / C_i.consider$

$i.2: [\mathbf{in}(C_i.Selected)] / C_i.activate$

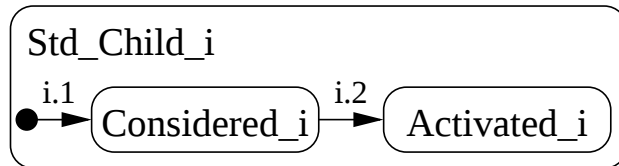


$i.3: [\mathbf{in}(C_i.Terminated)]$



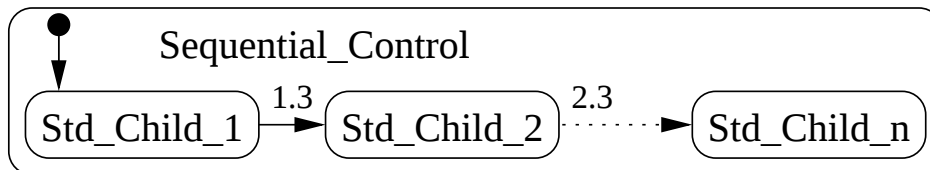
Plan Body (2)

Control Operators

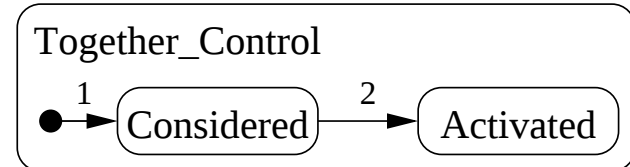


$i.1: / C_i.consider$

$i.2: [\mathbf{in}(C_i.Selected)] / C_i.activate$

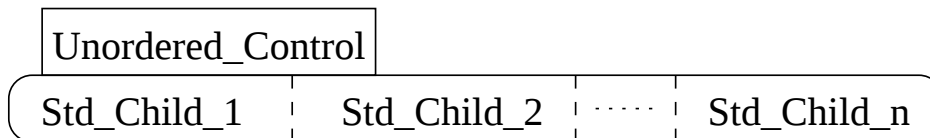


$i.3: [\mathbf{in}(C_i.Terminated)]$



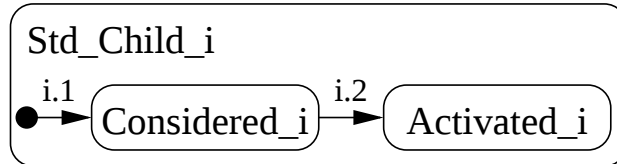
$1: / C_1.consider; \dots$

$2: [\wedge_{i=1}^k \mathbf{in}(C_i^m.Selected)]$
 $/ C_1.activate; \dots$



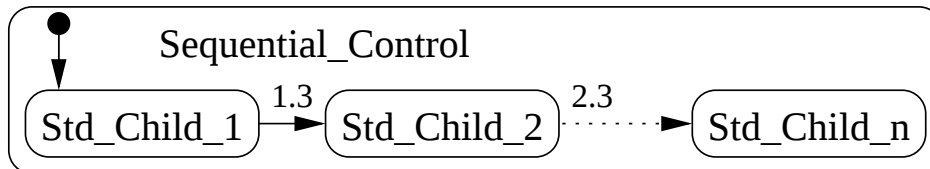
Plan Body (2)

Control Operators

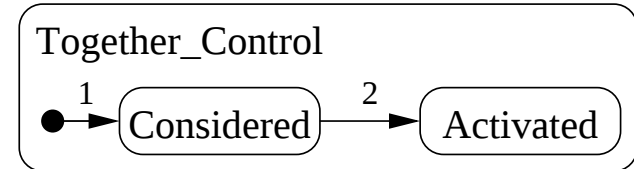
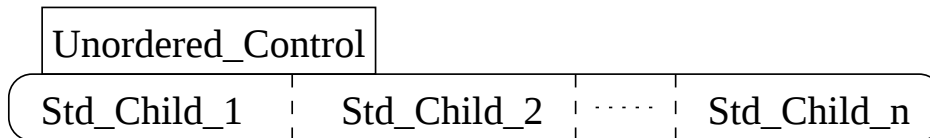


$i.1: / C_i.consider$

$i.2: [\mathbf{in}(C_i.Selected)] / C_i.activate$

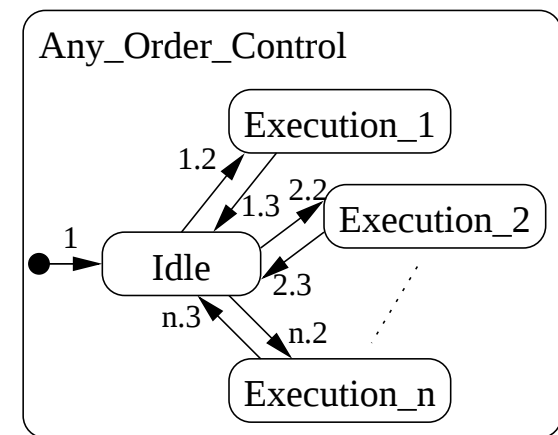


$i.3: [\mathbf{in}(C_i.Terminated)]$

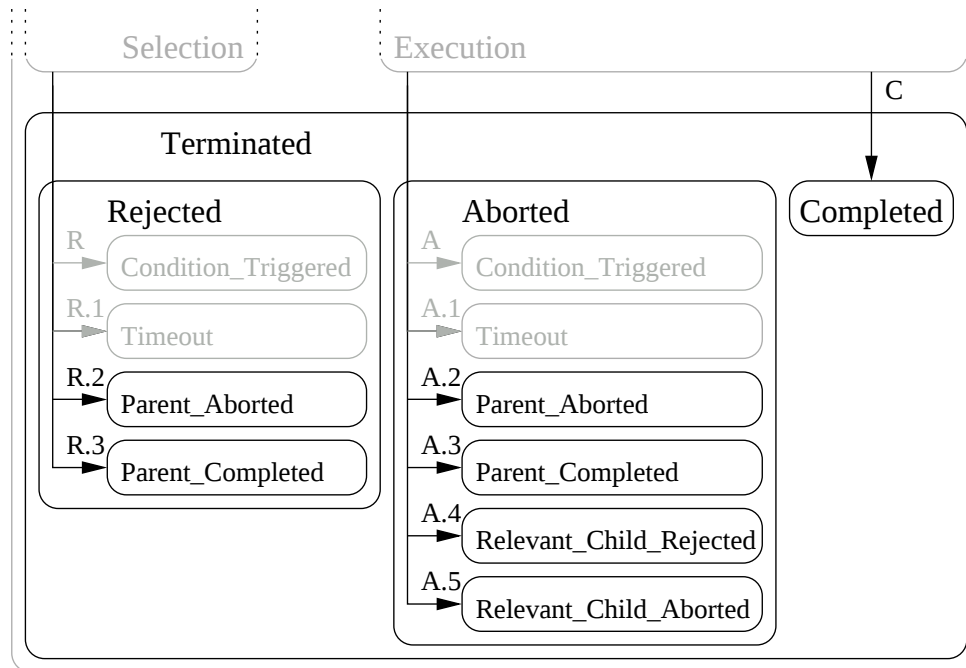


$1: / C_1.consider; \dots$

$2: [\wedge_{i=1}^k \mathbf{in}(C_i^m.Selected)]$
 $/ C_1.activate; \dots$



Additional synchronization of parent and sub plans
 $\Rightarrow$ **Propagation**



$R.2: [\mathbf{in}(P.Aborted)]$

$A.5: [\bigvee_{i=1}^k \mathbf{in}(C_i^m.Aborted)]$

$C : [\text{complete_cnd} = \dots]$

$\wedge \text{EFS} + \text{REF} \leq \dots$

$\wedge \text{MinDu} \leq \dots$

$\wedge \bigwedge_{i=1}^k \mathbf{in}(C_i^m.Completed)$

- propagation straightforward
- upward and downward propagation possible

Delay in Synchronization

- if parent aborts, child is signalled to also abort
- signalling child takes one micro step
- child can act in meantime

Delay in Synchronization

- if parent aborts, child is signalled to also abort
- signalling child takes one micro step
- child can act in meantime

Can this be tolerated?

Delay in Synchronization

- if parent aborts, child is signalled to also abort
- signalling child takes one micro step
- child can act in meantime

Can this be tolerated?

Three-Valued Conditions

- semantics requires three states true, false, unknown
- only reason is filter condition
- including setup condition would solve this problem

- precise semantics
- good documentation of Asbru language
- surprise: small picture for a large part of Asbru
- open problems
- additional effort necessary

Goal

Include semantics in deliverable

Todo

- refer to child plans as sub plans
- rename do-together to do-parallel
- support additional features

Support Which Features?

Basic Constructs

- intentions/effects

Conditions

- ? setup
- suspend/resume
- ? new style of activation

Data Handling

- ? context (data abstraction)
- local variables

Control Options

- do-cyclical
- do-one
- retry-aborted
- on-abort
- wait-for-optional