Formal models in industry standard tools: An Argos block within Simulink

IEHSC, May 2005, Singapore.

Timothy Bourke    Arcot Sowmya
School of Computer Science and Engineering
University of New South Wales, Sydney
and National ICT Australia
HYBRID = CONTINUOUS + DISCRETE
Stateflow

Many Statecharts features:

- hierarchy
- parallelism
- history junctions

Flowchart-like transitions:

- sequencing
- branching
- loops
Thinking/communicating about designs is involved:

1. intricate ordering rules
2. queued event processing
3. stacking of communications
4. implicit assumption of synchrony
Argos: a synchronous language

- Mealy machines
- Hierarchy
- Parallelism
- Discrete reactions
- Well-defined internal behaviour

Esterel, Lustre, Signal

CMA, INRIA, Verimag, IRISA

Argos is a synchronous version of Statecharts.

- Developed by Maraninchi and Rémond [Mar91, MR01].
- Well suited to some reactive programming tasks.
An Argos block: Syncblock

[CCM+03, SSC+04]
Syncblock Implementation
Syncblock Implementation

- Simulink
- S-function
- syncblock
Syncblock Implementation

- Simulink
- S-function
- Graphical editor
- syncblock
- runflows
- makeflows
- Argos program as text
- C source code
- FIFOs
- Boolean equations as text
- Interaction during simulation
Syncblock Implementation

Simulink

S-function

graphical editor

Argos program as text

makeflows

C source code

Boolean equations as text

FIFOs

interaction during simulation
Syncblock Implementation

Simulink

S-function

syncblock

c

makeflows

runflows

Argos program as text

graphical editor

Boolean equations as text
Syncblock Implementation

- Graphical editor
- Argos program as text
- Makeflows
- Runflows
- Boolean equations as text
- C source code
- Simulink
- S-function
- Syncblock
- Compile (CC)
Syncblock Implementation

Graphical editor

Simulink

S-function

syncblock

FIFOs

program

makeflows

runflows

Argos program as text

Boolean equations as text

C source code

CC
Syncblock Implementation

- **Simulink**
- **S-function**
- **graphical editor**
- **makeflows**
- **runflows**
- **Argos program as text**
- **FIFOs**
- **syncblock**
- **program**
- **interaction during simulation**
- **CC**

- **Boolean equations as text**
- **C source code**
Compilation [MH96]

\[
\begin{align*}
\text{power} & = (\text{one} \land \text{switch}) \\
\text{none}’ & = (\text{one} \land \text{timeout} \land \neg \text{switch}) \lor (\text{one} \land \text{switch}) \\
& \quad \lor (\text{none} \land \neg \text{switch}) \\
\text{one}’ & = (\text{none} \land \text{switch}) \lor (\text{one} \land \neg \text{timeout} \land \neg \text{switch}) \\
\text{off}’ & = (\text{off} \land \neg \text{power}) \lor (\text{on} \land \text{power}) \\
\text{on}’ & = (\text{on} \land \neg \text{power}) \lor (\text{off} \land \text{power})
\end{align*}
\]
Example: Fault-Tolerant Fuel Control System [Mat]
Ego in range = FAIL

O2_normal

O2_fail

O2_warmup

O2_cold, FAIL_O2

[48 TSEC]

O2_normalentry: fail_state[O2] = 0;

entry: fail_state[O2] = 1;

O2_failentry: fail_state[O2] = 1;

[Sens_Failure_Counter.DEC]

[t > o2_t_thresh]

[Sens_Failure_Counter.INC]

Oxygen_Sensor_Mode
Summary

- Stateflow is powerful but has shortcomings
- Existing research might help
- Argos block developed:
  - contrast with Stateflow
  - simple examples possible
  - paucity of features has pros and cons

Need the right tool for the task at hand.
References


[MR01] Florence Maraninchi and Yann Rémond. Argos: an automaton-based synchronous