

Demonstrating interoperability of FORTE runtime with synchronous execution environment

Valeriy Vyatkin

v.vyatkin@auckland.ac.nz

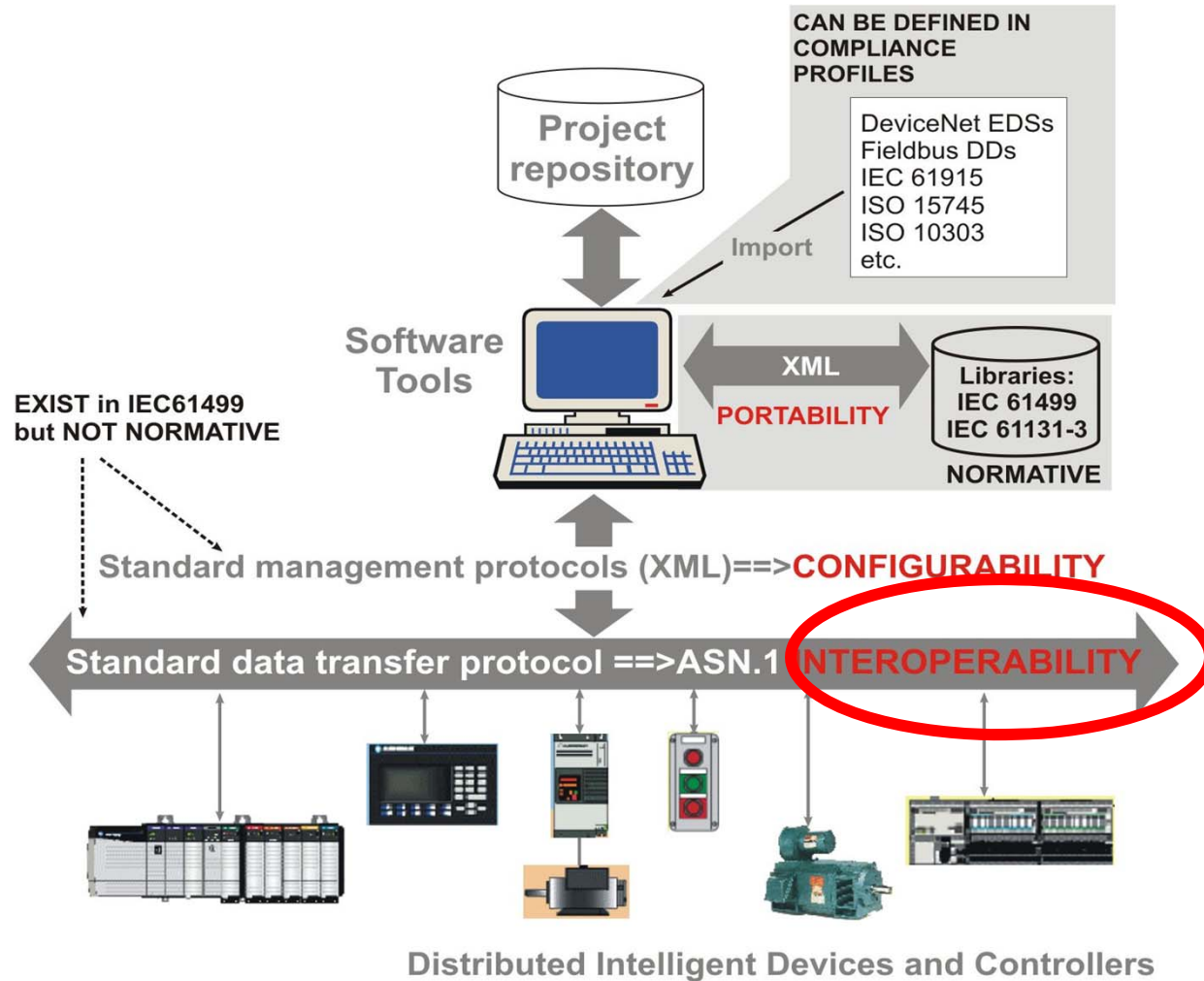
Majid Sorouri (Presenter)

The University of Auckland, NZ

Plan

- Interoperability in IEC 61499
- Synchronous execution of function blocks
- BlokSynk – synchronous compiler
- BlokSynk plugin to NxtStudio
- Interoperability demo

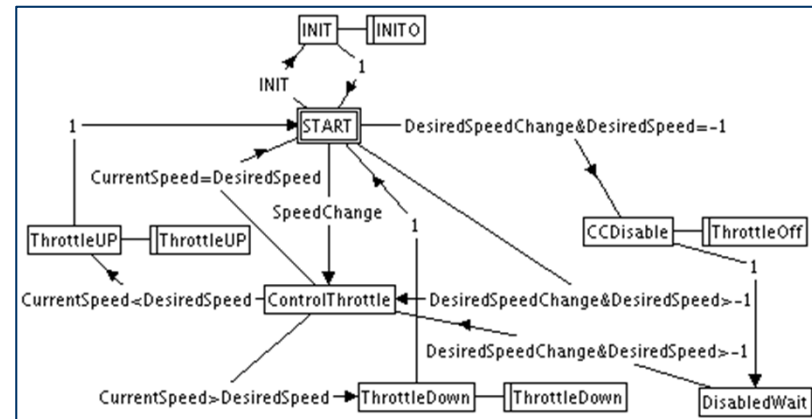
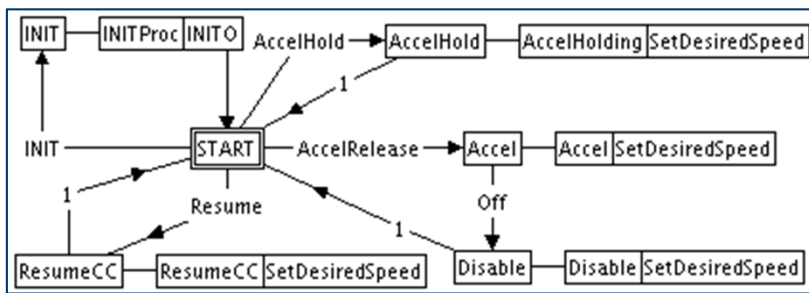
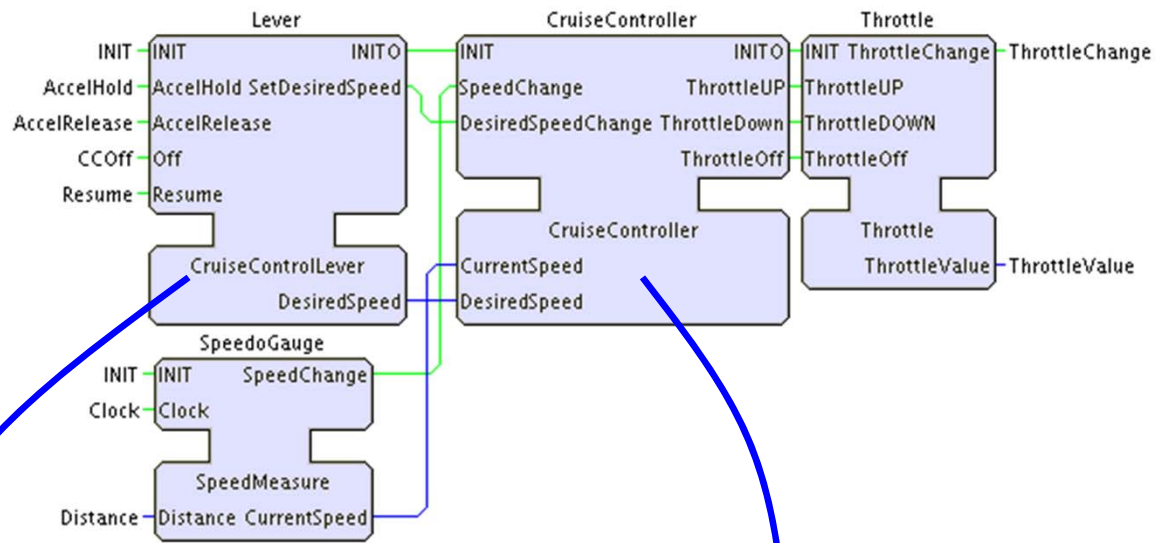
Interoperability



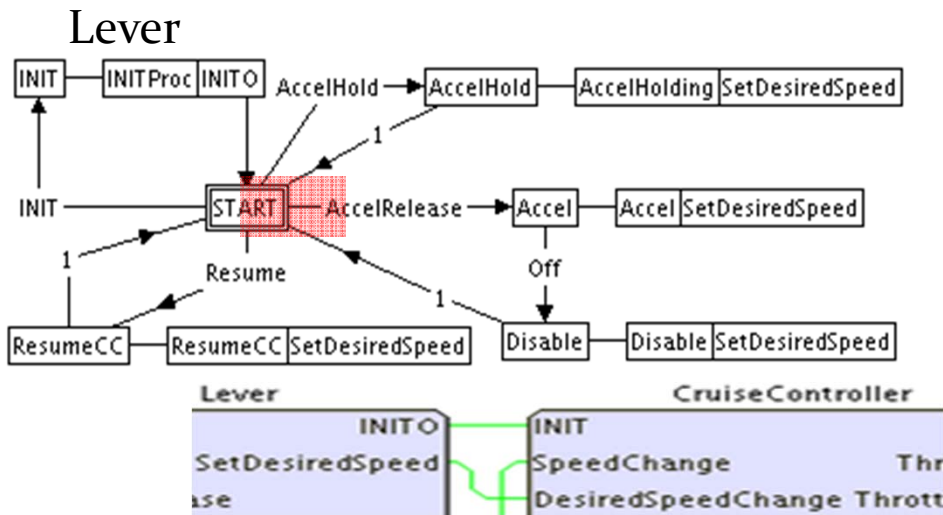
Other provisions of interoperability?

- Generic communication models, e.g.:
 - PUBLISH-SUBSCRIBE
 - CLIENT-SERVER
- Event-driven, so good chance of being tolerant to different execution models of IEC 61499

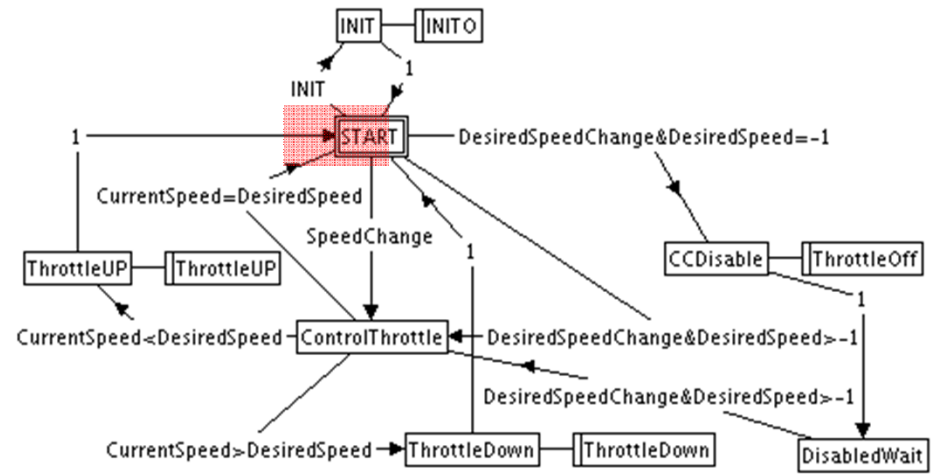
FB Synchronous Execution Semantics



Synchronous Execution

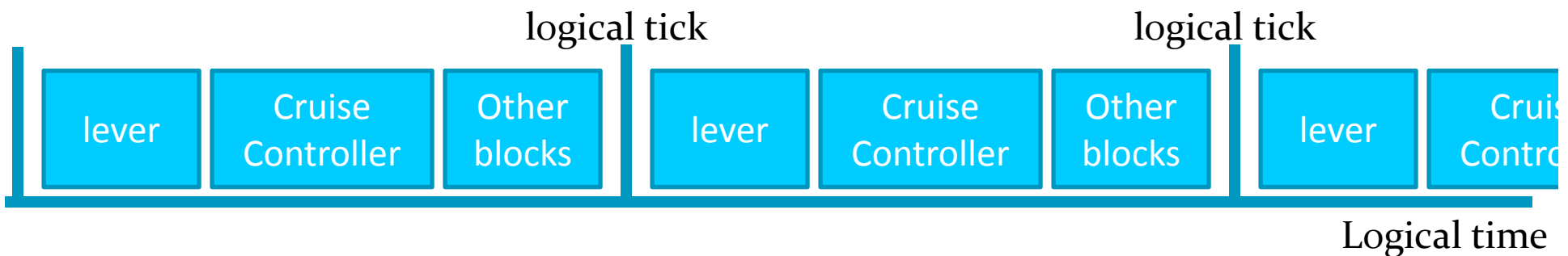


Cruise Controller

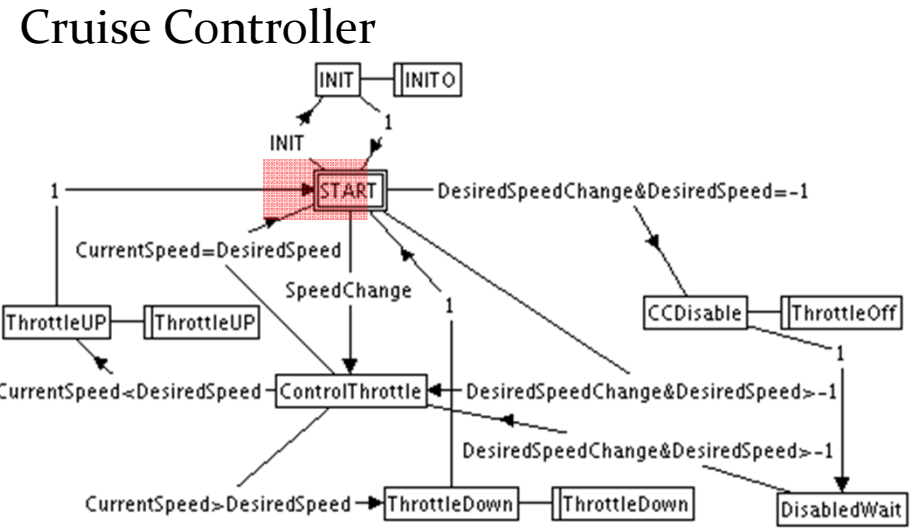
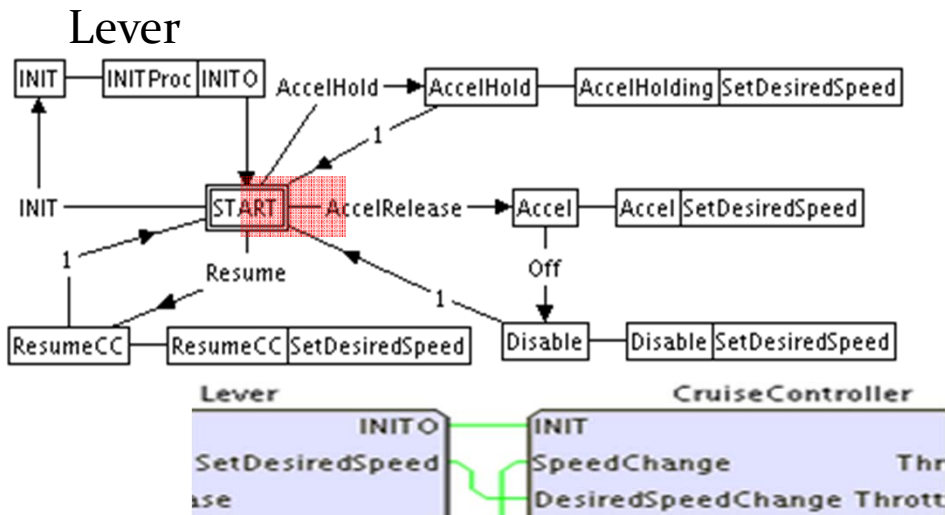


Inputs events:

Outputs events:

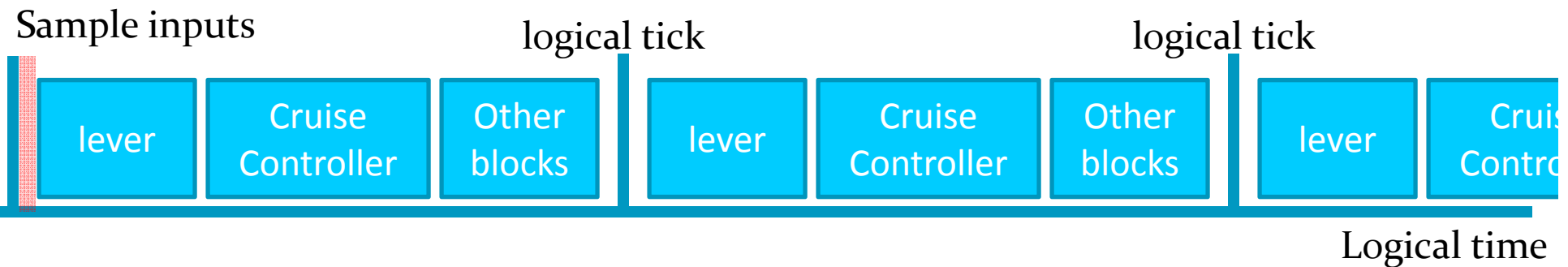


Synchronous Execution

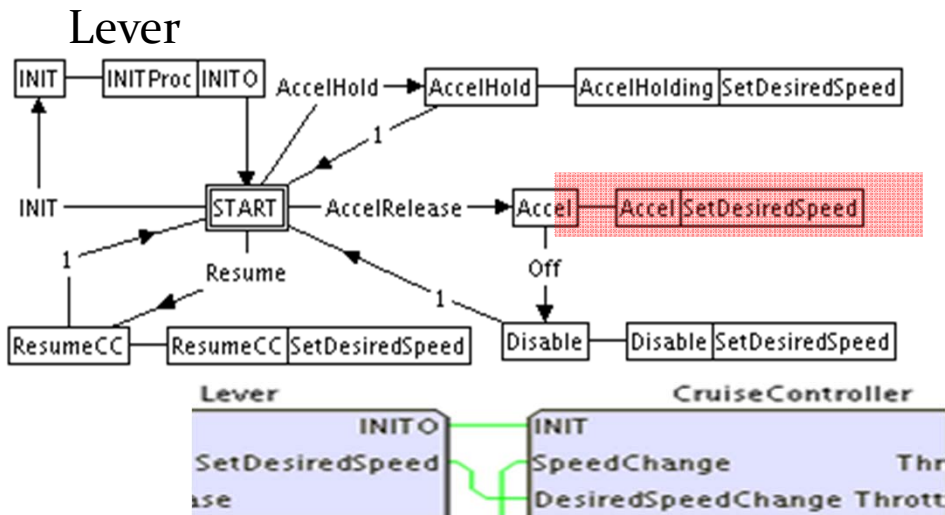


Inputs events:
AccelRelease

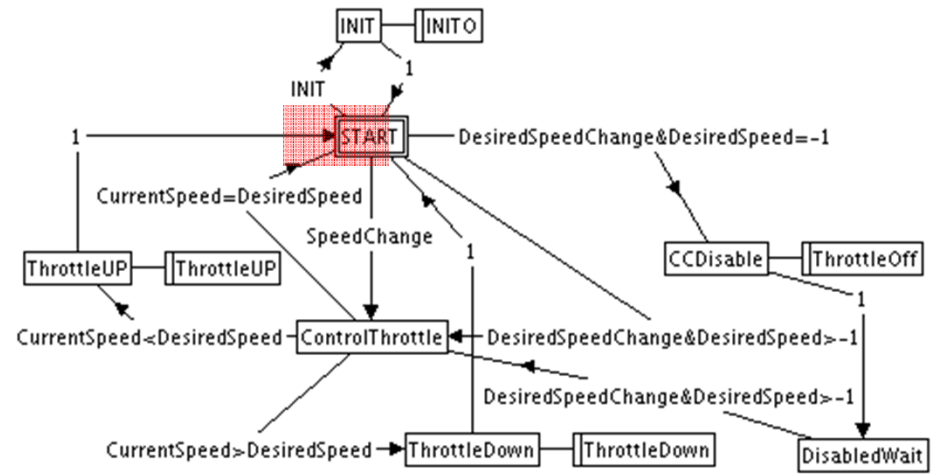
Outputs events:



Synchronous Execution

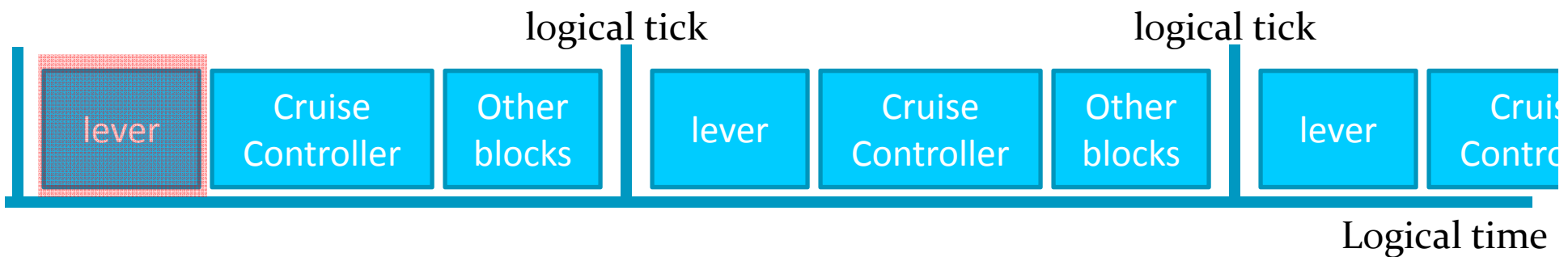


Cruise Controller

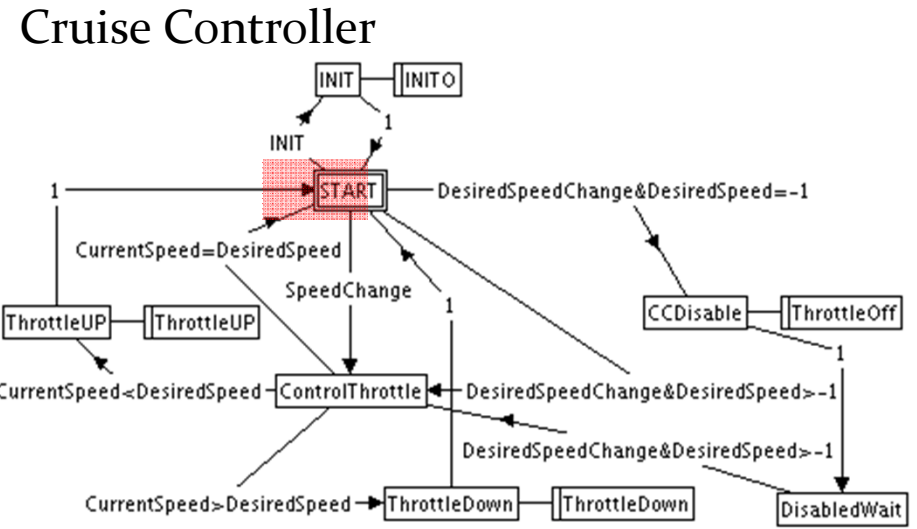
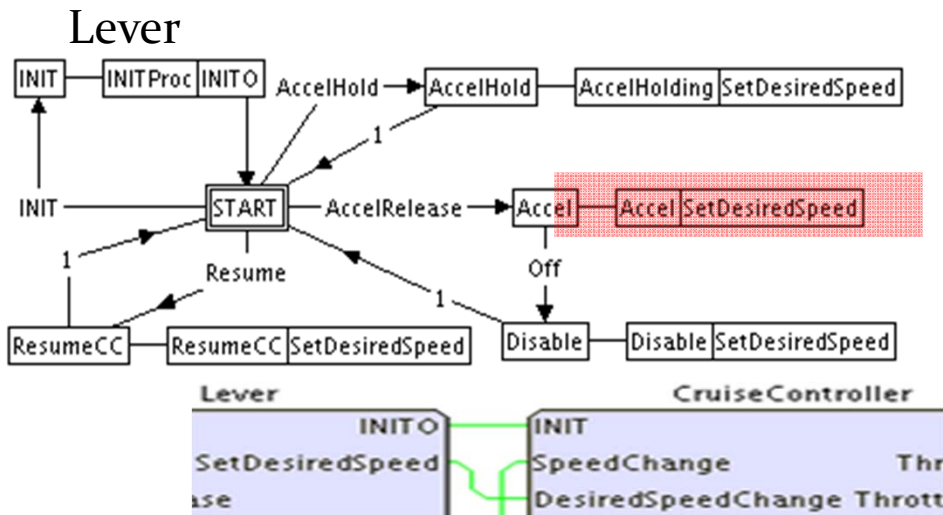


Inputs events:
AccelRelease

Outputs events:

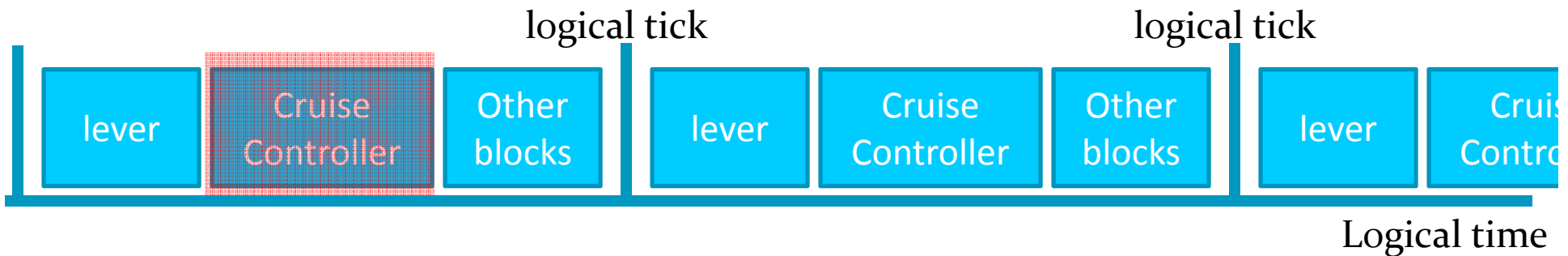


Synchronous Execution

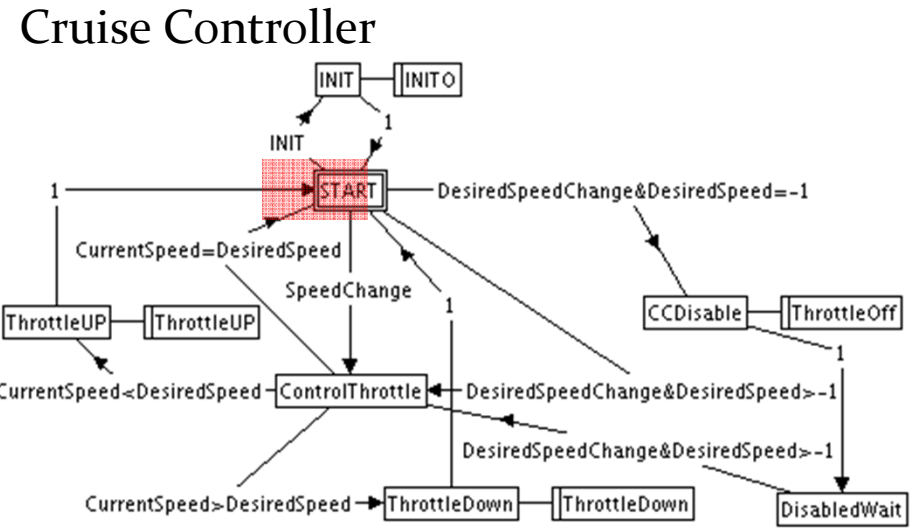
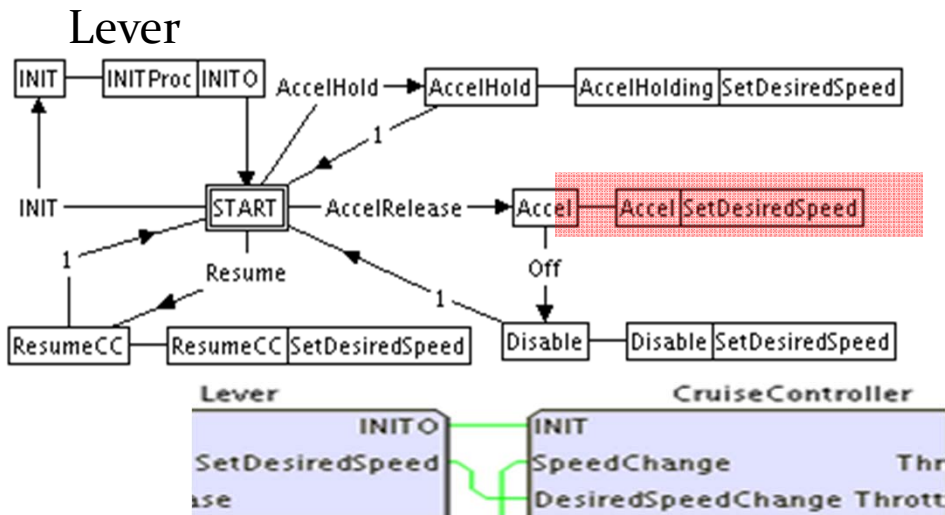


Inputs events:
AccelRelease

Outputs events:

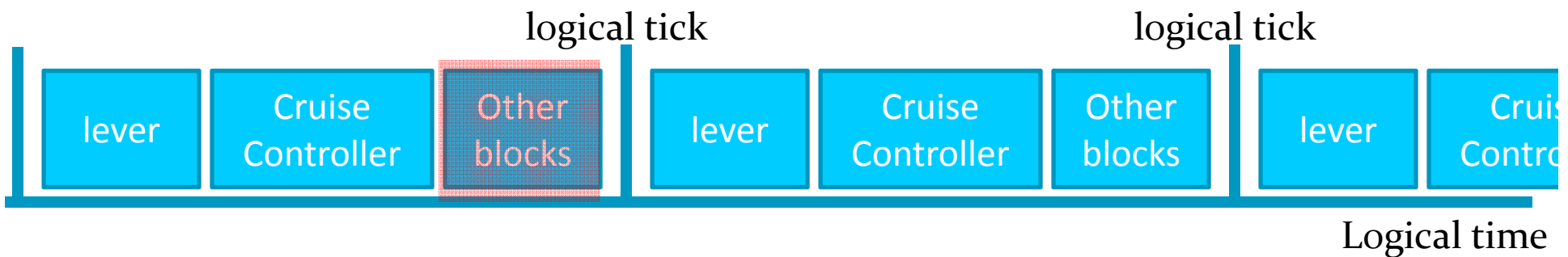


Synchronous Execution

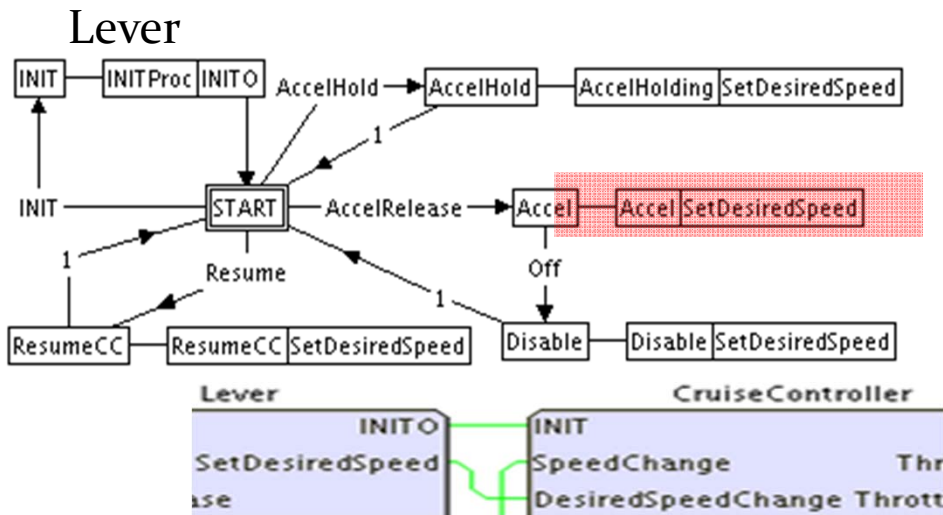


Inputs events:
AccelRelease

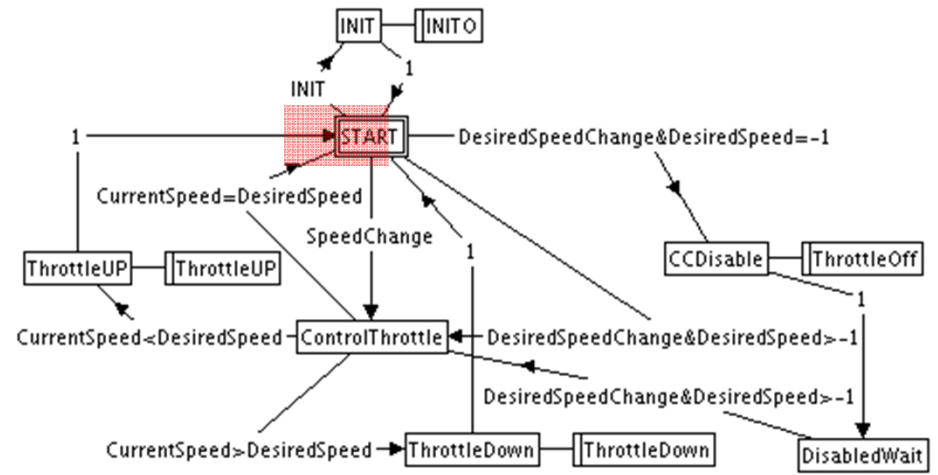
Outputs events:



Synchronous Execution

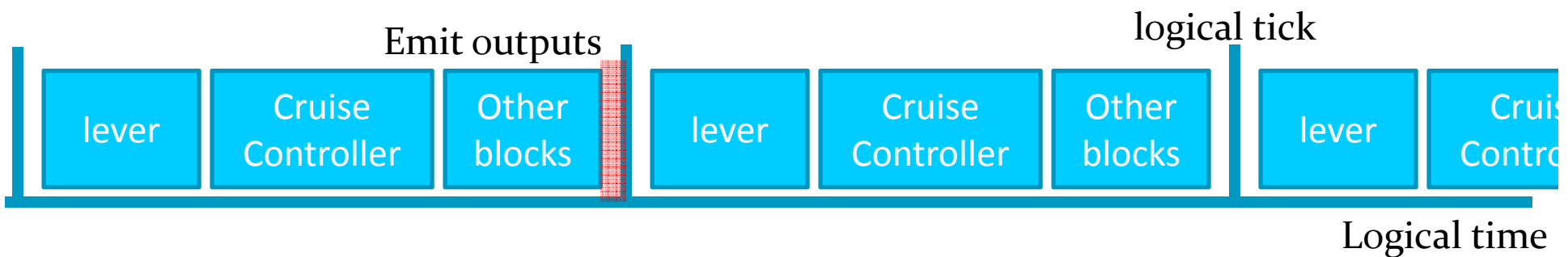


Cruise Controller

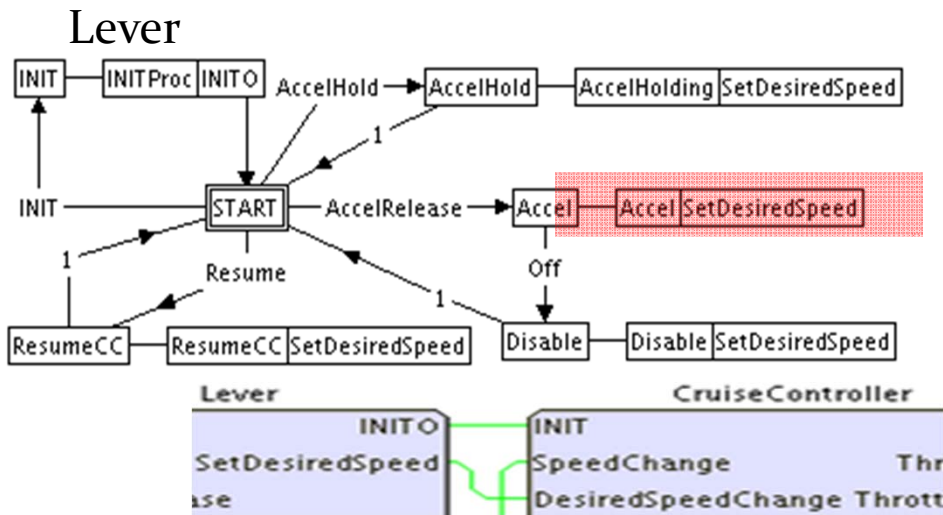


Inputs events:
AccelRelease

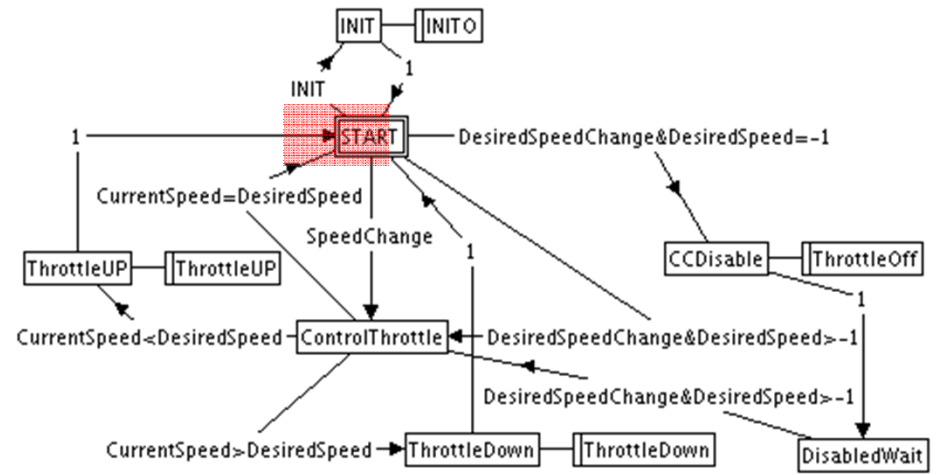
Outputs events:
SetDesiredSpeed



Synchronous Execution



Cruise Controller

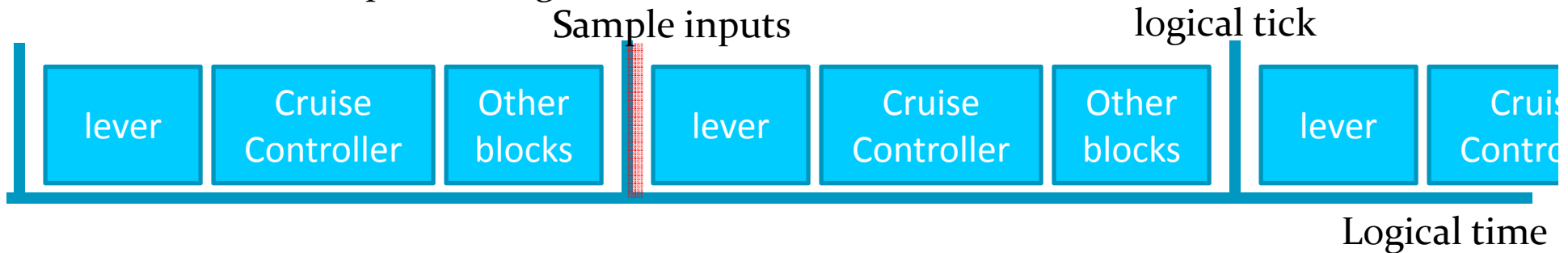


Inputs events:

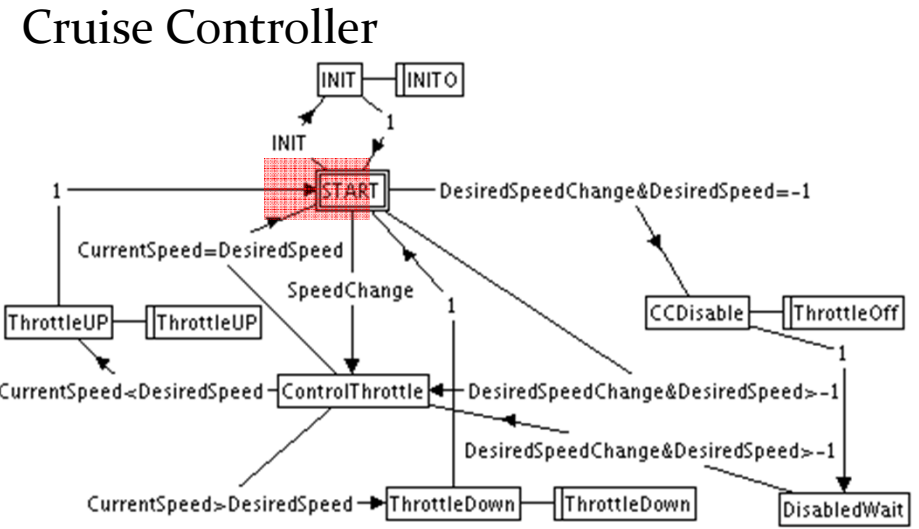
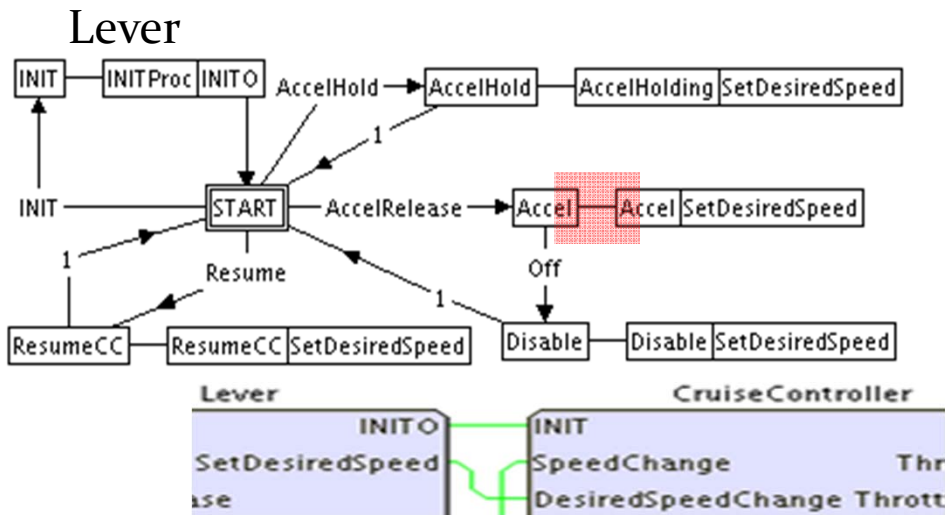
- AccelRelease
- DesiredSpeedChange

Outputs events:

- SetDesiredSpeed



Synchronous Execution



Inputs events:

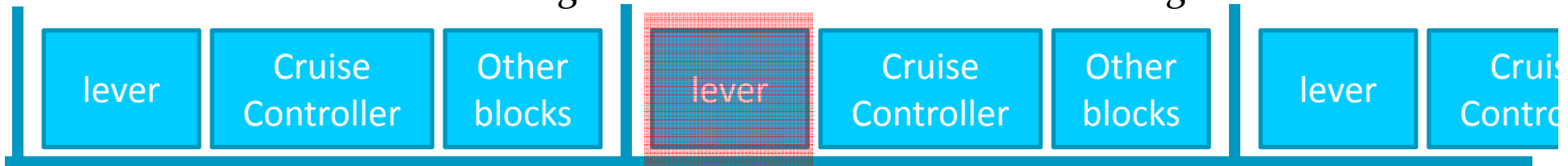
AccelRelease
DesiredSpeedChange

Outputs events:

SetDesiredSpeed

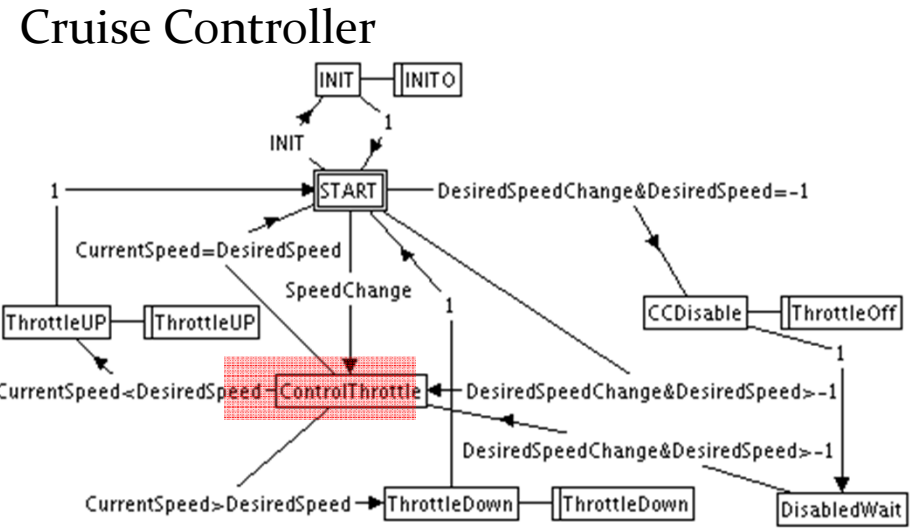
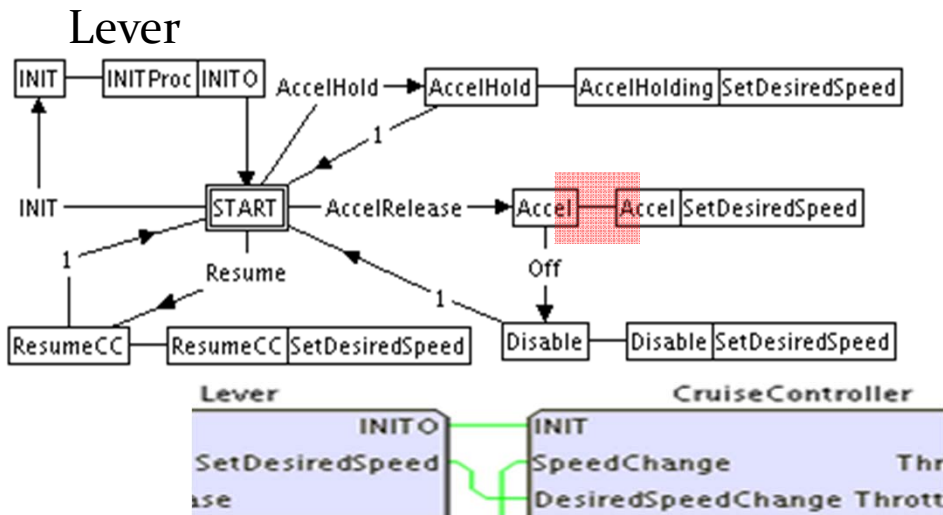
logical tick

logical tick



Logical time

Synchronous Execution



Inputs events:

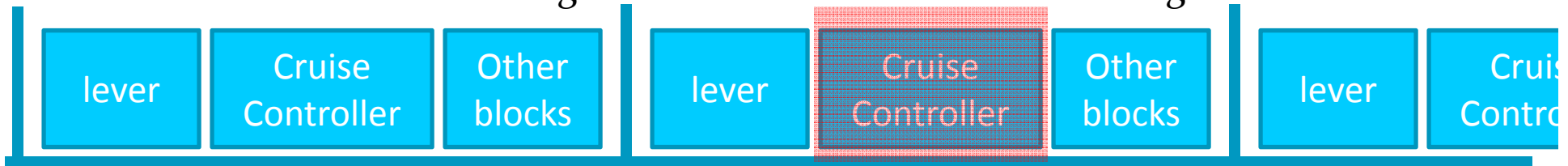
AccelRelease
DesiredSpeedChange

Outputs events:

SetDesiredSpeed

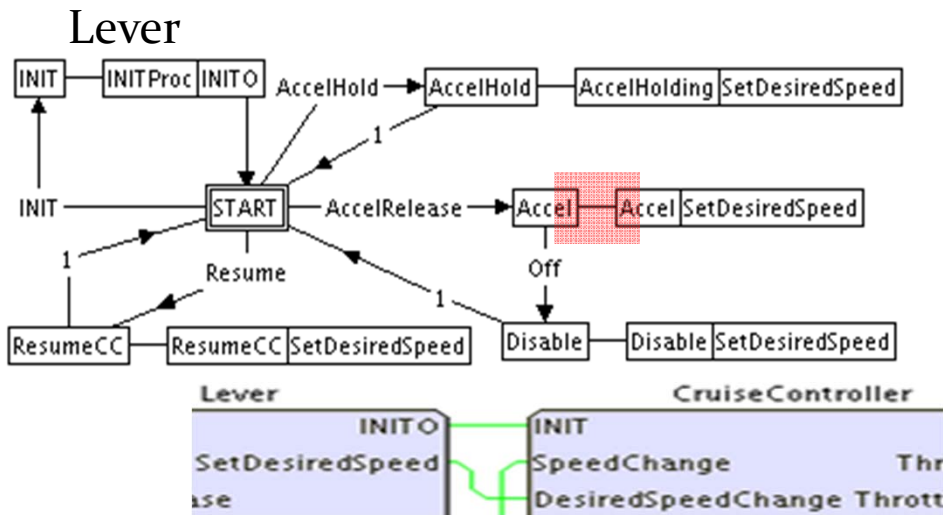
logical tick

logical tick

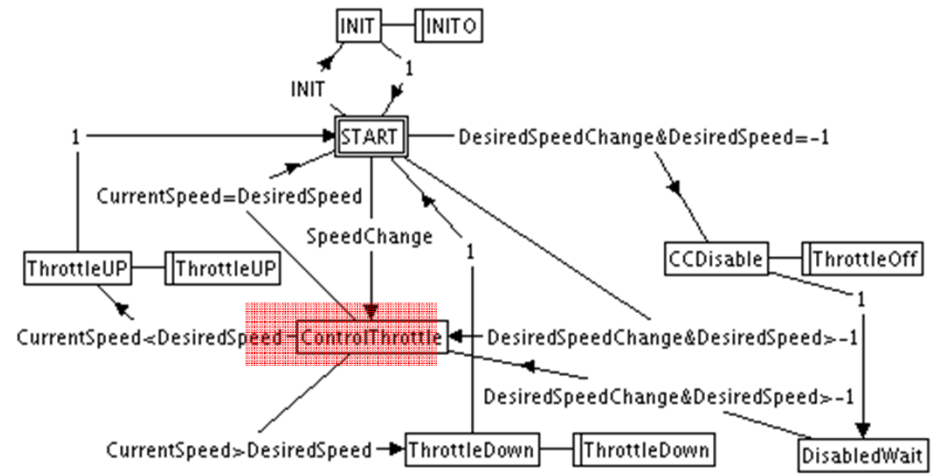


Logical time

Synchronous Execution



Cruise Controller



Inputs events:

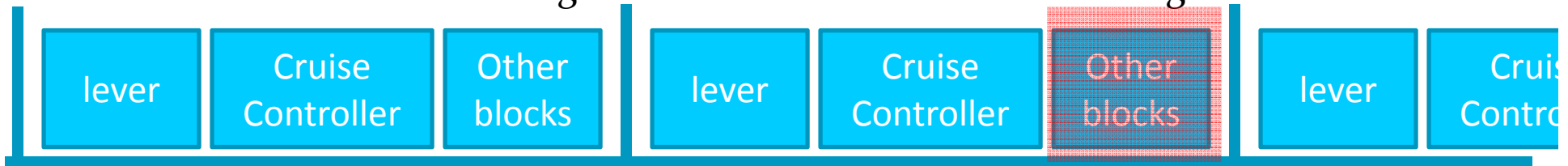
AccelRelease
DesiredSpeedChange

Outputs events:

SetDesiredSpeed

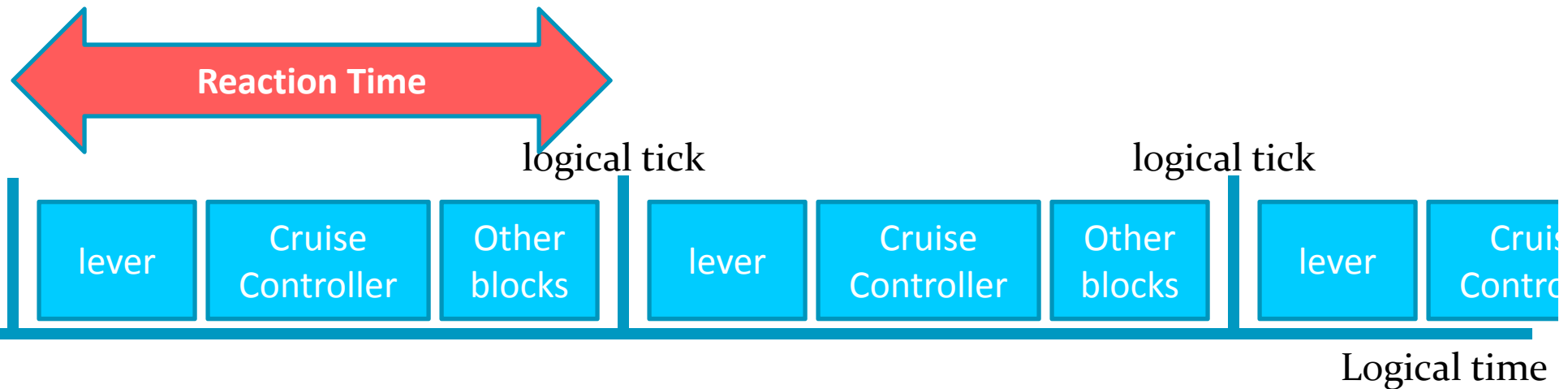
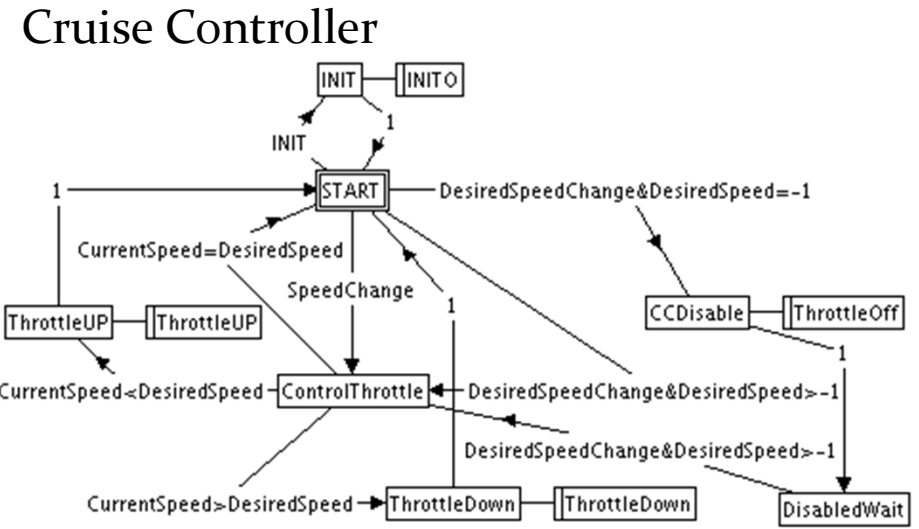
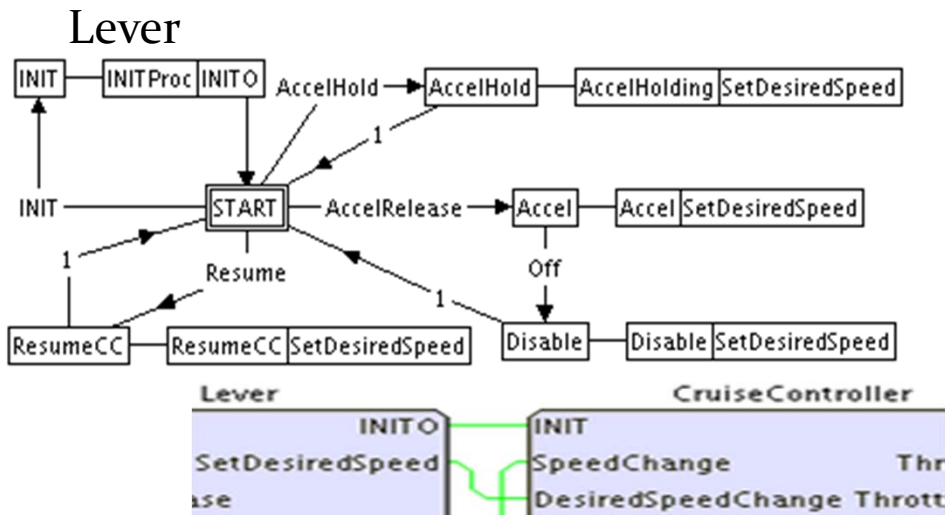
logical tick

logical tick

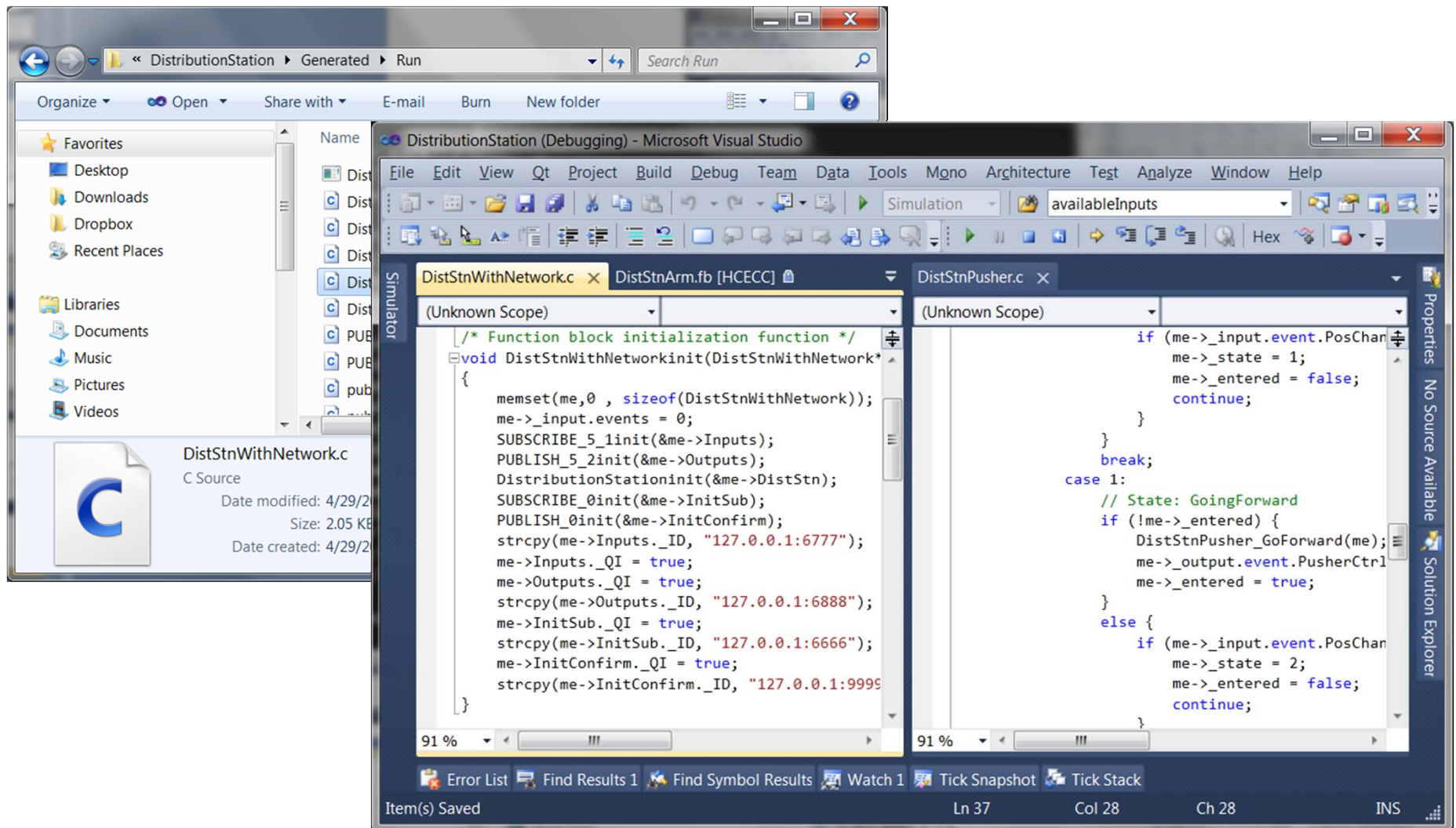


Logical time

Synchronous Execution



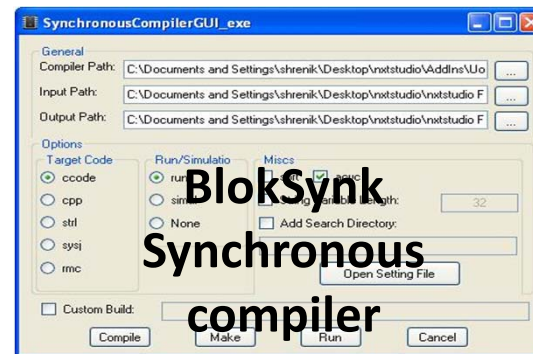
BlokSynk: Generation of Deterministic Safe C code



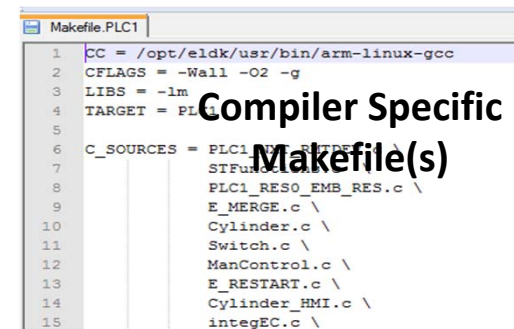
BlokSynk Plugin



FB Model in
NxtStudio

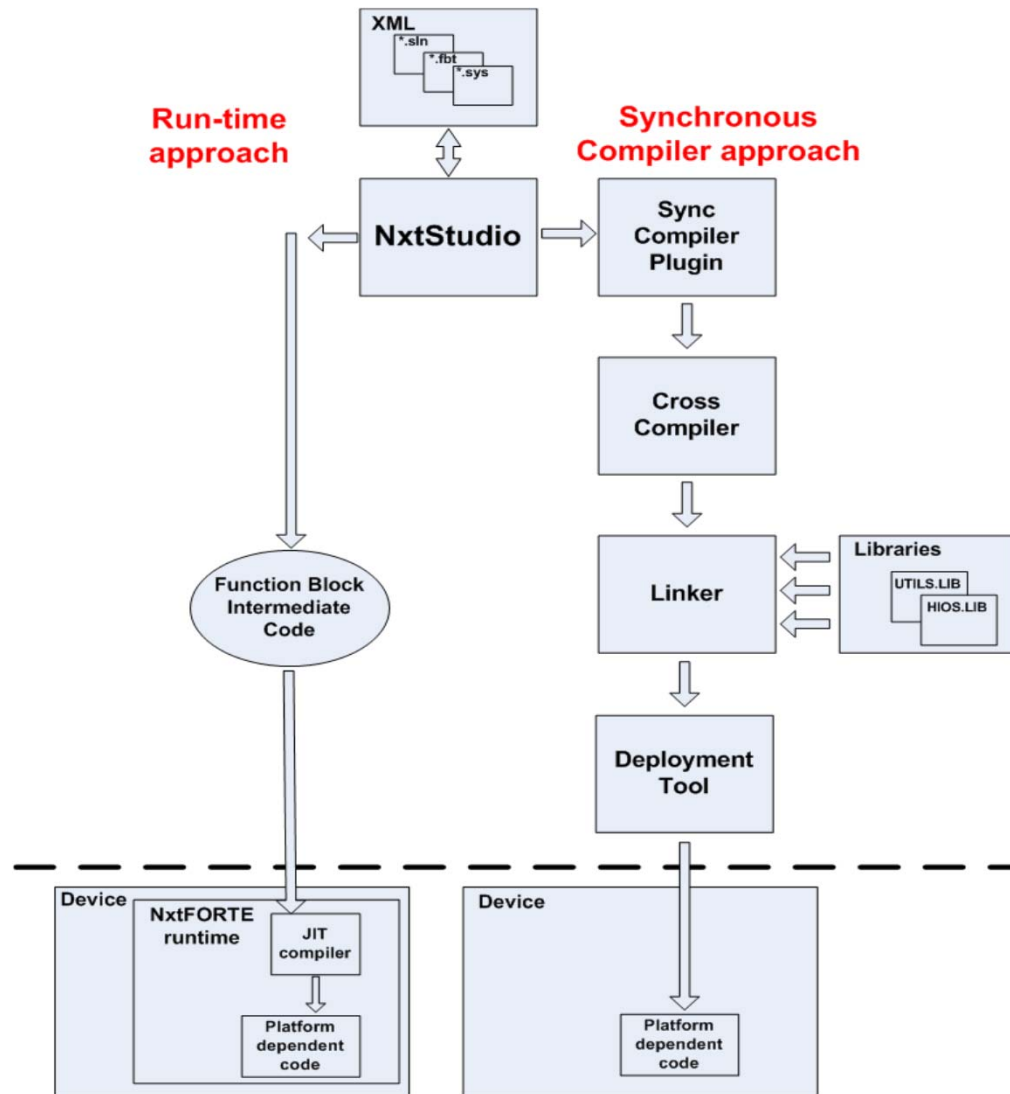


Standard 'C'
source Files

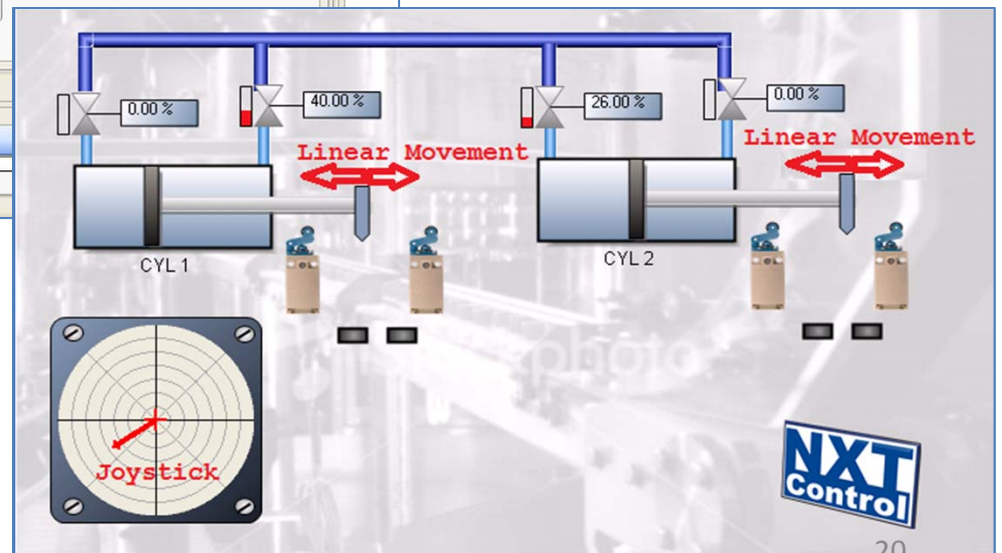
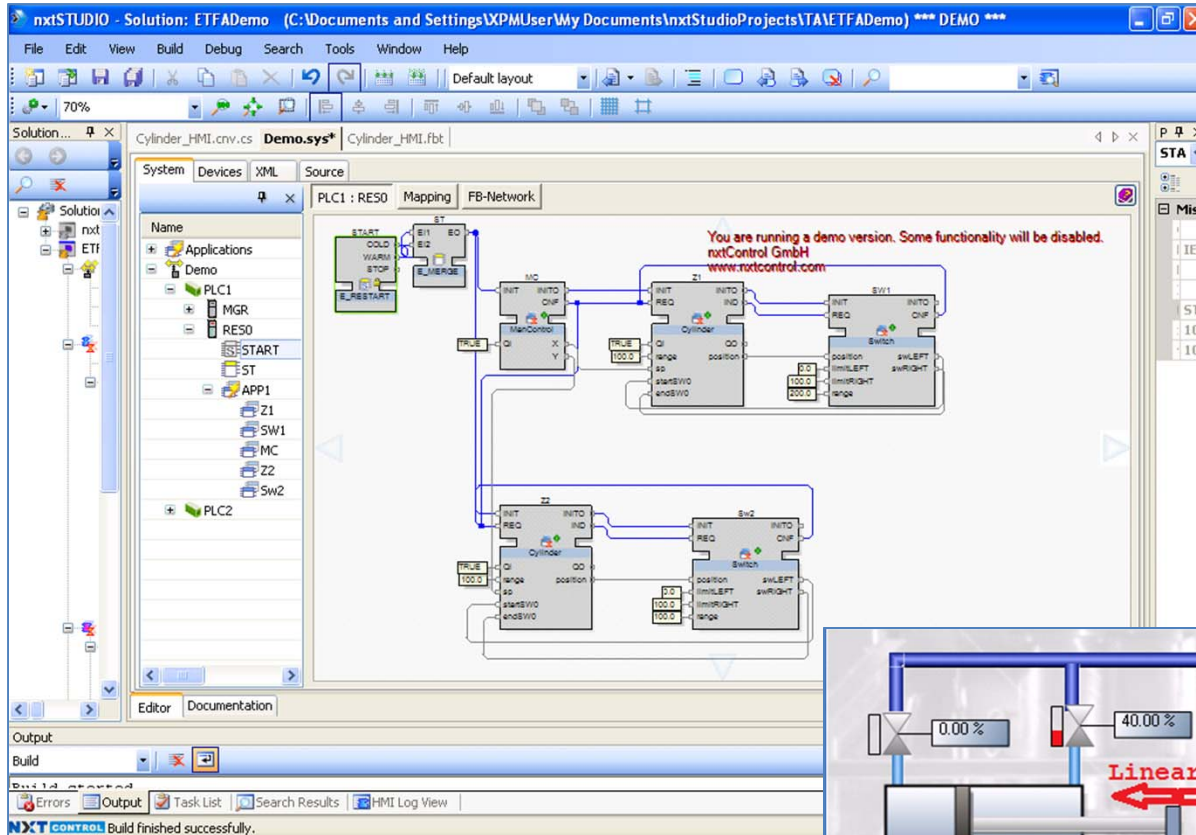


Compiler Specific
Makefile(s)

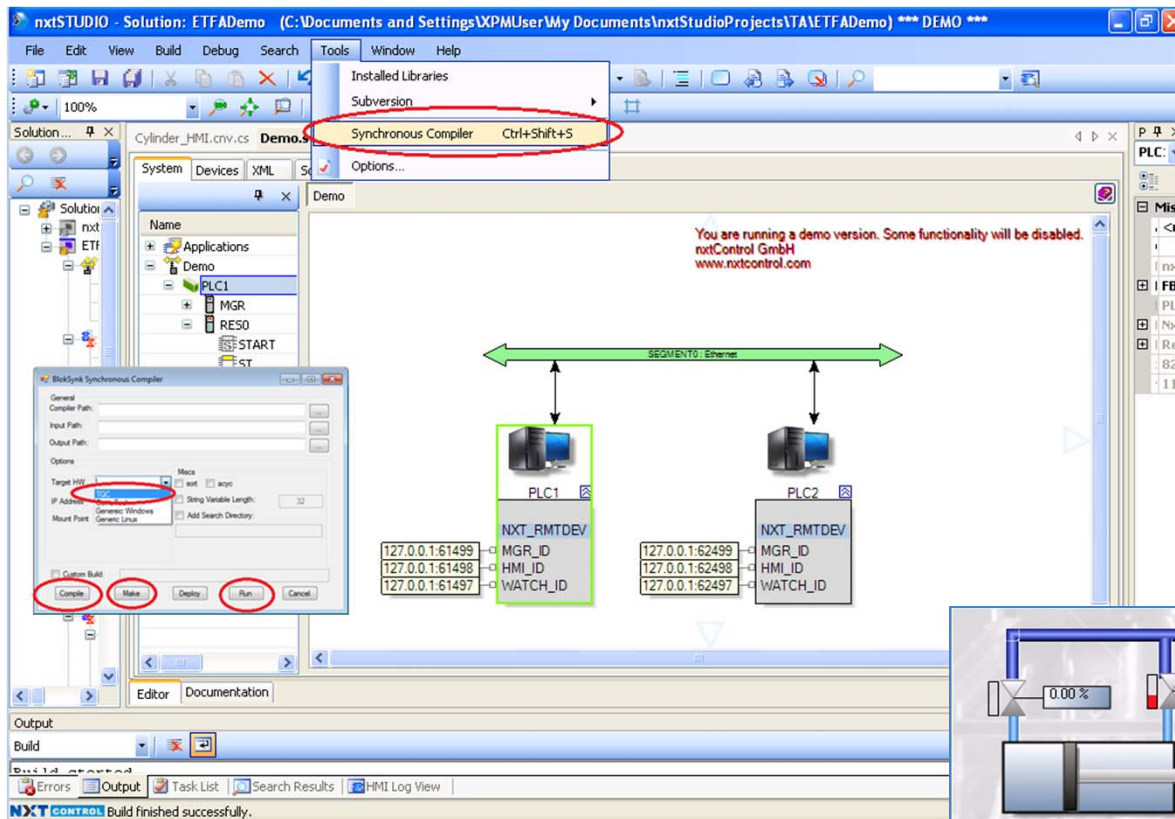
BlokSynk Deployment Model



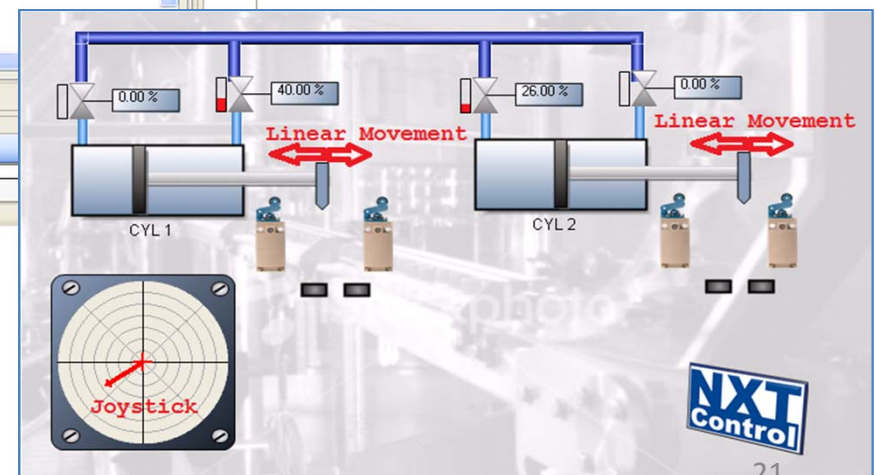
Demo



Deployment view



The screenshot shows the nxtSTUDIO interface for a project named 'ETFDemo'. The 'Tools' menu is open, and the 'Synchronous Compiler' option is highlighted with a red circle. Below the menu, the 'Synchronous Compiler' dialog box is visible, with the 'Run' button also circled in red. The main workspace shows a network diagram with two PLCs (PLC1 and PLC2) connected via an Ethernet segment. Each PLC is associated with an 'NXT_RMTDEV' device and has three IP addresses listed: MGR_ID, HMI_ID, and WATCH_ID. The output window at the bottom displays the message: 'Build Finished successfully.'



Substituting Device On the Fly

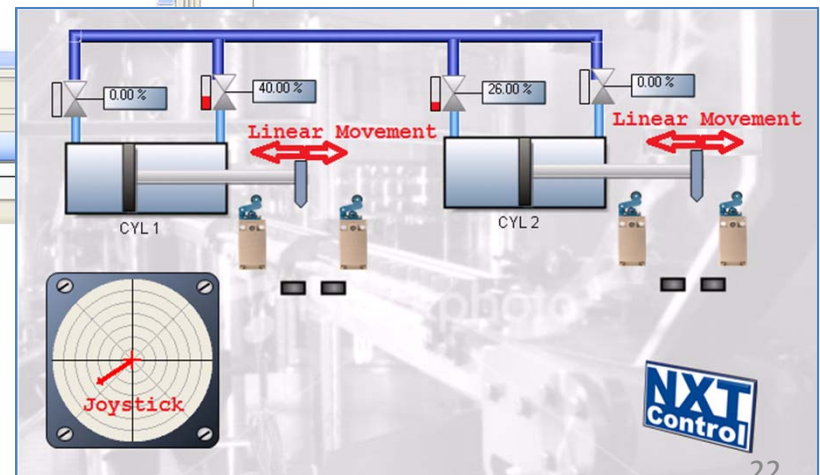
START

The screenshot shows the nxT STUDIO interface. The 'Tools' menu is open, highlighting the 'Synchronous Compiler' option. Below the menu, a network diagram shows two PLCs, PLC1 and PLC2, connected via an Ethernet segment. PLC1 is associated with IP addresses 127.0.0.1:61499, 127.0.0.1:61498, and 127.0.0.1:61497. PLC2 is associated with 127.0.0.1:62499, 127.0.0.1:62498, and 127.0.0.1:62497. A red 'X' is drawn over the diagram, and a red arrow points from the 'START' text to the diagram.

```

File Edit Setup Control Window Help
dramsetupdump fw printenv hello lcdtuner Pictures sram-test
root@tqma28:~# cd /home/nxt/
root@tqma28:/home/nxt# ls
Makefile Makefile.PL1 Makefile.PL2
Cylinder.c Cylinder.h Cylinder.HMI.c Cylinder.HMI.h
Cylinder.o Cylinder.o.old Cylinder.HMI.o Cylinder.HMI.o.old
ManControl.c ManControl.h ManControl.HMI.c ManControl.HMI.h
ManControl.o ManControl.o.old ManControl.HMI.o ManControl.HMI.o.old
NxtCommunication.c NxtCommunication.h NxtCommunication.o NxtCommunication.o.old
NxtHMIData.c NxtHMIData.h NxtHMIData.o NxtHMIData.o.old
pubsub.c pubsub.h pubsub.o pubsub.o.old
NxtHMIServer.c NxtHMIServer.o NxtHMIServer.o.old
NxtHMIServerConfig.c NxtHMIServerConfig.h NxtHMIServerConfig.o NxtHMIServerConfig.o.old
NxtRMTDEV.c NxtRMTDEV.h NxtRMTDEV.o NxtRMTDEV.o.old
SplitCoord.h SplitCoord.o
STFunctions.h STFunctions.o
Switch.h Switch.o
Switch.HMI.c Switch.HMI.h
Switch.EMB.RES.c Switch.EMB.RES.h
Valve.c Valve.o
Valve.HMI.c Valve.HMI.h
Valve.EMB.RES.c Valve.EMB.RES.h
Valve.EMB.RES.HMI.c Valve.EMB.RES.HMI.h
Valve.EMB.RES.HMI.o Valve.EMB.RES.HMI.o.old
Valve.o Valve.o.old
root@tqma28:/home/nxt# ./PLC1 & Run PLC1 in
[1] 1610
root@tqma28:/home/nxt# NxtHMIServer listening on port: 61498.
root@tqma28:/home/nxt# ./PLC2 & Run PLC2 in
[2] 1611
root@tqma28:/home/nxt# NxtHMIServer listening on port: 62498.
root@tqma28:/home/nxt#
    
```

STOP



Conclusion

- BlokSynk and Forte interoperate with each other
- BlokSynk talks to Nxt HMI
- This is yet another benefit of IEC 61499!