Fringe Session:
Developing JIWY using TERRA
Communicating Process Architectures 2012

Maarten Bezemer, Robert Wilterdink and Jan Broenink
Robotics and Mechatronics, University of Twente, The Netherlands
Summary of the Paper

- CSP Meta-Model just presented

- Used by
  - CSP Editor
  - Model validation
  - Code generation (FDR, C++/LUNA)

- Obviously it all is very nice (and works)

- At least that is what we just told...
Time for a demo!

- CSP Editor
  Modeling a simple robotic setup

- Transformation to CSPm/FDR

- Transformation to C++/LUNA

- Execute C++ code on target
CSP Editor

- JIWY with an architecture model
  - Two controllers (Pan & Tilt)
  - Safety / Scaling model
  - Interaction with hardware

- CSP Models to implement the sub-models
  - Communication defined by the architecture model

- Not available unfortunately

- So we will use only a CSP model
CSP Editor

- JIWY with a CSP model
  - Controller sub-models
    - Readers/Writers for data communication
    - Empty sub-models used for (20-sim) controller algorithm containers
  - Safety sub-model
    - Readers/Writers for data communication
    - Separated for Pan and Tilt signal
    - Empty sub-models used to add some C++ code for safety checks
  - Fake interaction with hardware
    - Thrown away!
    - Channels replaced by link drivers to communicate with HW
Live demo
Transformation to CSPm / FDR

- Formal verification of TERRA models
- Basic CSP objects can be transformed
- C++ Code (blocks) not... (obviously)
  - Process = SKIP
- Formally checking of robotic oriented models is limited
  - Only software structure (ie pure CSP)
Live demo
Transformation to C++ / LUNA

- All CSP constructs represented by C++ / LUNA code
- All (sub-)models expressed in header/source file
  - Contains channels, processes, ports, groups, etc.
- Modifying generated code is possible
  - Protected regions: comment, constructor/destructor, execute()
- Code blocks filled in by custom code
  - Or by code generated from control law design tools (e.g. 20-sim)
- Additional generated files
  - Entry point (main() function)
  - Makefile to build the application
Live demo
Execute on target

- Compile & Link
  - Should be automated in the future

- Send executable to (QNX) target
  - Should be automated in the future

- Execute, experiment, test, ...!
  - (and hope we did not make any mistakes...)
  - Would be nice to automate as well, but probably stays manual...
Execute on target

Live demo
That's all folks

- As claimed by the paper presentation
  
  Usable for “Developing Embedded Control Software”

- If you are still interested
  
  Come to us for a 'Do It Yourself' session!