Integrating synthesis, verification, co-simulation, visualization and code generation for supervisory control of complex industrial systems

Bert van Beek (D.A. van Beek)
Systems Engineering Group
Eindhoven University of Technology
• Airport baggage handling system: large scale distributed and cooperative system
  • State of the art development process
  • Two layer control system design: routing layer + ECS layer
  • Challenges in control system design of the ECS layer
  • Research project: ECS control layer design using CIF

• Compositional Interchange Format (CIF)
  • Overview
  • Tooling
  • Theoretical foundation
  • Large scale system design support

• Demonstrations and videos on CIF controller design
  • Validation by means of interactive simulation and visualization
  • Videos of real-time control of an industrial 46 conveyor belt system using generated PLC code
Two layer control system design
• Scheduling and routing layer: many different implementation platforms
• Equipment Control Software (ECS) layer: PLCs
Large scale distributed and cooperative system

• Equipment Control Software distributed over 100+ PLCs
• Each PLC can control 100+ sections (e.g. conveyors)
• Guaranteeing high Quality of Service essential
• Transportation of a wide range of baggage inevitably leads to errors
• All errors need to be efficiently and reliably handled
• Low level failures propagate through all control levels and may lead to baggage rerouting
• Operator intervention often necessary => stringent safety requirements on equipment and control systems
State of the art in ECS layer control design

• Initial design estimates by means of discrete-event stochastic performance analysis models

• For large baggage handling systems
  • all requirements (informally) specified in database
  • each requirement is covered by a separate test case
  • commissioning by rigorous testing on actual system
  • difficult to relate informal requirements to low level PLC code

• Hardware-in-the-loop simulation by means of hardware models on PCs

• Several back ends
  • Airports generally require PLCs (Siemens, Allen Bradley, CodeSys)
    • For systems with high speed accelerations also e.g. C

• Porting systems to different back ends leads to recoding
CIF for baggage handling control system design

- Model of control system
- Model of physical uncontrolled system
- Validation by means of interactive simulation and visualization
- Generate real-time code for different back ends by means of property preserving model transformations

Why CIF?
CIF screenshot

Eclipse: cross-platform open source IDE

Examples browser

Editor with “as you type” syntax checker

Plotting

SVG interactive visualization

State visualizer for debugging

Console for textual output

Examples:
- Browser editor with “as you type” syntax checker
- SVG interactive visualization
- Plotting
- State visualizer for debugging
- Console for textual output
Characteristics of the Compositional Interchange Format (CIF)

- Based on hybrid automata
- For supervisory control of hybrid, timed and untimed systems
- Strong theoretical foundation: formal compositional semantics
- Large set of open source tools, based on open standards
- Tool integration via model transformations and cosimulation
- Open source via MIT license

Development of CIF supported by several EU (and National) projects, including:

- CIF 1: HYCON, Darwin
- CIF 2: C4C, MULTIFORM, HYCON2, Darwin
- CIF 3: HYCON2
CIF 3 documentation: 200+ webpages

CIF 3

Welcome to the website for CIF 3, the Compositional Interchange Format for hybrid systems. CIF automata-based modeling language for the specification of discrete event, timed, and hybrid systems. The CIF tooling supports the entire development process of controllers, including specification, supervisory controller synthesis, simulation-based validation and visualization, verification, real-time testing, code generation, etc.

CIF 3 was created and is currently developed by the Systems Engineering group of the Mechanical Engineering department, at the Eindhoven University of Technology (TU/e).

The CIF 3 tooling is free, and is available under the MIT open source license.

Information

About   Download   Support   Index   Search

Documentation

Language tutorial   CIF 3 textual syntax (PDF)   Event-based supervisory controller   Tools   Changelog for version r7363

CIF 3 text editor

Supervisory controller synthesis

- Event-based synthesis toolset
- CIF to Supremica transformer
- CIF to SCIM transformer

Simulation, validation, and visualization

- CIF 3 simulator
- CIF 3 to CIF 2 transformer

Verification

- CIF to mCRL2 transformer
- CIF to Supremica transformer

Real-time testing, code generation, and implementation

- CIF 3 PLC code generator

Miscellaneous

- CIF to CIF transformer
- CIF merger
- CIF 3 event disabler
CIF: Theoretical foundation

• Much simplified formal compositional operational semantics
• The meaning of any CIF component is defined independently of its environment
• Bisimulation *proven* to be a congruence for all operators
• Compositionality essential for transformations:

  If the transformation $T(C)$ of a controller $C$
  is equivalent (bisimilar) to $C$, then
  
  $T(C) || P$
  is equivalent to
  
  $C || P$

  for all CIF components $P$
CIF large scale systems design support

• Parameterized process definition and instantiation (templates)
  • for models and for visualizations
• Grouping of components for hierarchical modeling in separate scopes
• Complementary synchronization and communication concepts
  • Multi-party synchronizing events well known from automata; additional partners *restrict* behavior (AND functionality)
  • Point to point communication via channels; additional partners *relax* behavior (OR functionality)
  • Unique in CIF 3: events synchronize with channels (observers/supervisors)
• Shared variables: local write, global read
CIF tool integration

Cosimulation via S-function in Matlab / Simulink

Implementation: Uni Dortmund / Christian Sonntag
CIF tool integration

- Cosimulation
  - Matlab / Simulink
  - Modelica
  - Many other tools
- External function calls via Java
- Large number of property preserving CIF to CIF model transformations
- Several transformations to other tools
  - supervisory control synthesis
  - verification
  - code generation
CIF controller design and implementation

Small baggage handling system example:
- CIF controller and hardware model
- Validation by interactive simulation
- Manual conversion to PLC code
Conveyor physical model and controller visualization

Virtual positions in controller

Photo Electric Cell

Conveyor

Add/Remove products for interactive simulation

Switch on/off for interactive simulation
Simulation with visualization

- SWS status: retracted
- TRS_1 status: stopped
- TRS_2 status: starting up
- TRS_3 status: starting up
- TRS_4 status: energy save
- TRS_5 status: energy save

State visualizer:

- Time: 27.5
- SWS: VB

Control panels:
- TRZ 1: Start, Stop, Reset, Pause
- TRZ 2: Start, Stop, Reset, Pause
- VBZ_3: Start, Stop, Mode, Switch
  - Auto: true, Dir: straight
- MGZ_4: Start, Stop, Mode, Switch
  - Auto: true, Dir: side
- Area 1: Start, Stop
Concluding remarks

• Control system for large conveyor system consisting of about 600 automata successfully converted to PLC real-time control system

• Current and future work
  • Extension of functionality of control system
  • Improving readability and efficiency of code execution
  • Validation of correctness of transformation

• CIF open source ([cif.se.wtb.tue.nl](http://cif.se.wtb.tue.nl)):
  • Code can be freely downloaded, used and adapted
  • For co-development of CIF: contact Dennis Hendriks