Business from technology

Ontology-Driven Natural Language Requirement Templates for Model Checking I&C Functions

EHPG 2013

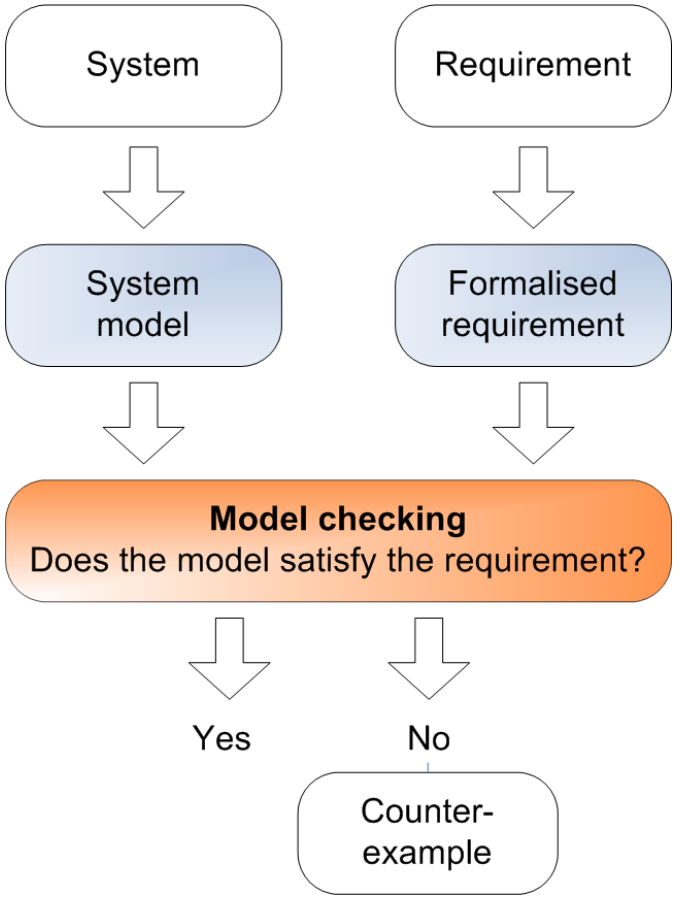
Teemu Tommila, Antti Pakonen, Janne Valkonen
VTT Technical Research Centre of Finland

Controlled Natural Language (CNL) in requirements definition

- Motivation
  - Poor quality of textual requirements (vagueness, complexity of sentences, inexact terminology)
  - Need for formalised requirements in advanced tools, e.g. model checking
  - Existing research and technologies in the fields of CNL and semantic web
- The approach
  - Develop recommended natural language templates together with corresponding formal representations
  - Define a vocabulary (ontology) for the domain
  - Develop a concept of a “syntax and ontology controlled requirements editor”

Model checking

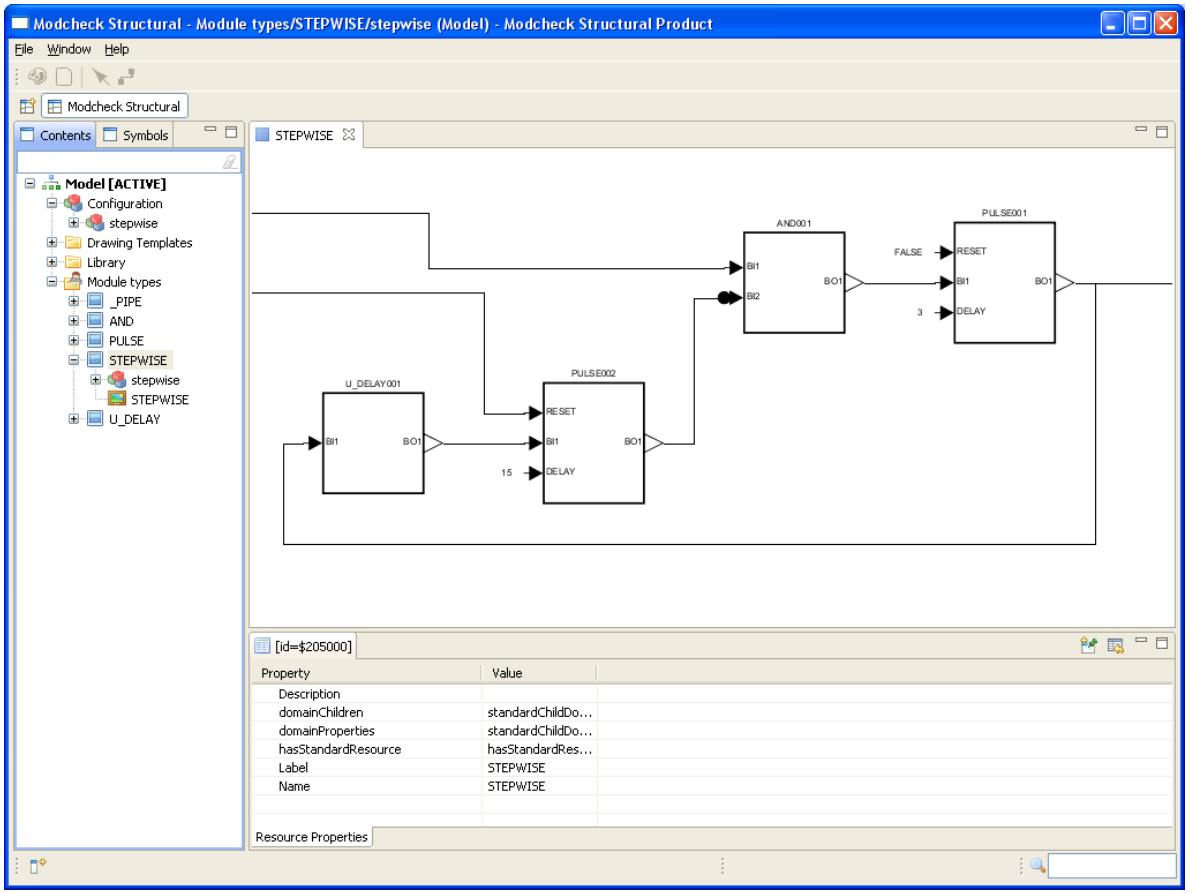
- Model checking is an efficient formal method for the verification of critical systems.
- Model checker is a software tool that exhaustively checks that a model of a system satisfies given requirements.
  - The answer is either **YES**…
  - …or **NO**, in which case the model checker will output a ***counter-example***, demonstrating an execution path that does not meet the requirement.



2D modelling with Simantics

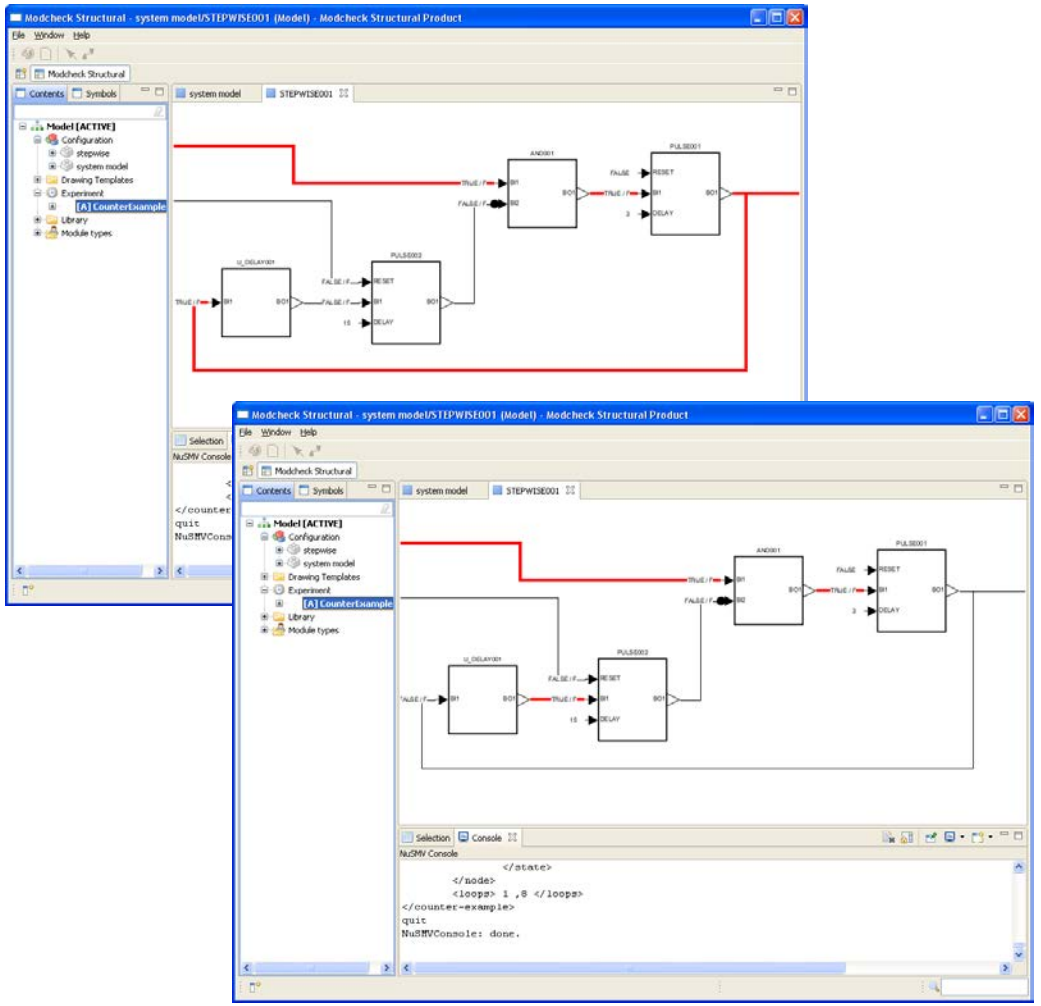
- Simantics – open-source platform for modelling and simulation
- https://www.simantics.org/

- Specify the function block diagram in a 2D view.



Browsing counter-examples with Simantics

- Review of counter-examples with ”living” function block diagrams
- In the future: automatic transformation of SMV model from design data



Requirement templates

- Mapping of natural language description with the associated temporal logic
- ”Fill in the blanks” – refer to domain concepts (i.e. devices, functions, I&C system inputs and outputs)

Natural language template	Temporal logic representation
P occurs at most 2 times between **Q** and **R**	G((**Q** & F(**R**)) -> ((!**P** & !**R**) U (**R** | ((**P** & !**R**) U (**R** | ((!**P** & !**R**) U (**R** | ((**P** & !**R**) U (**R** | (!**P** U **R**))))))))))

Requirement templates – an example

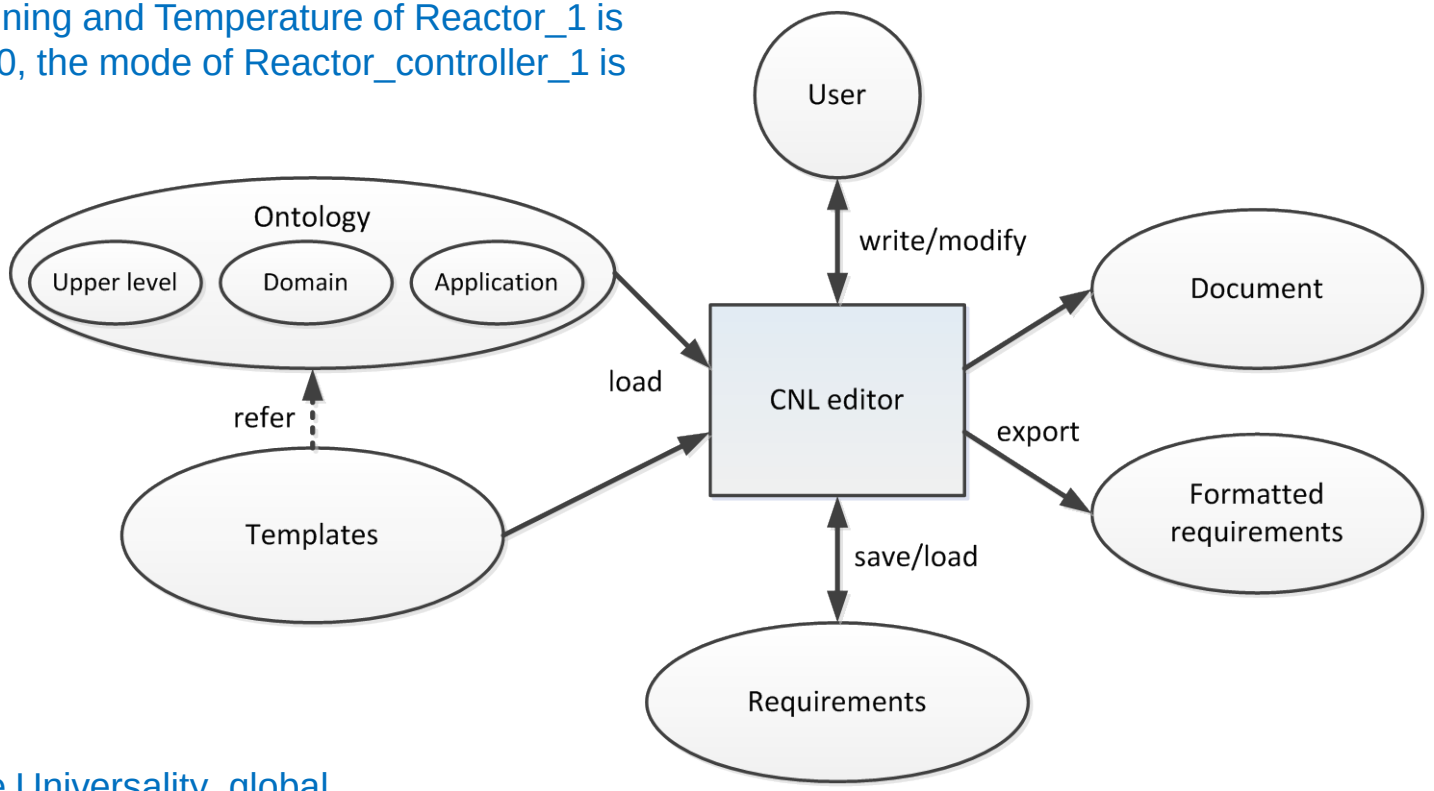
Requirement:

After the shutdown signal, valve V15 must be opened and remain open until the level in tank T4 is below 230 cm.

Natural language template	Temporal logic representation
V15_OPEN must hold after SHUTDOWN_SIGNAL until (T4_LEVEL_M < 230)	G((SHUTDOWN_SIGNAL & !((T4_LEVEL_M < 230))) ->(G(V15_OPEN) | ! V15_OPEN U (T4_LEVEL_M < 230))))

CNL editor, the overall concept

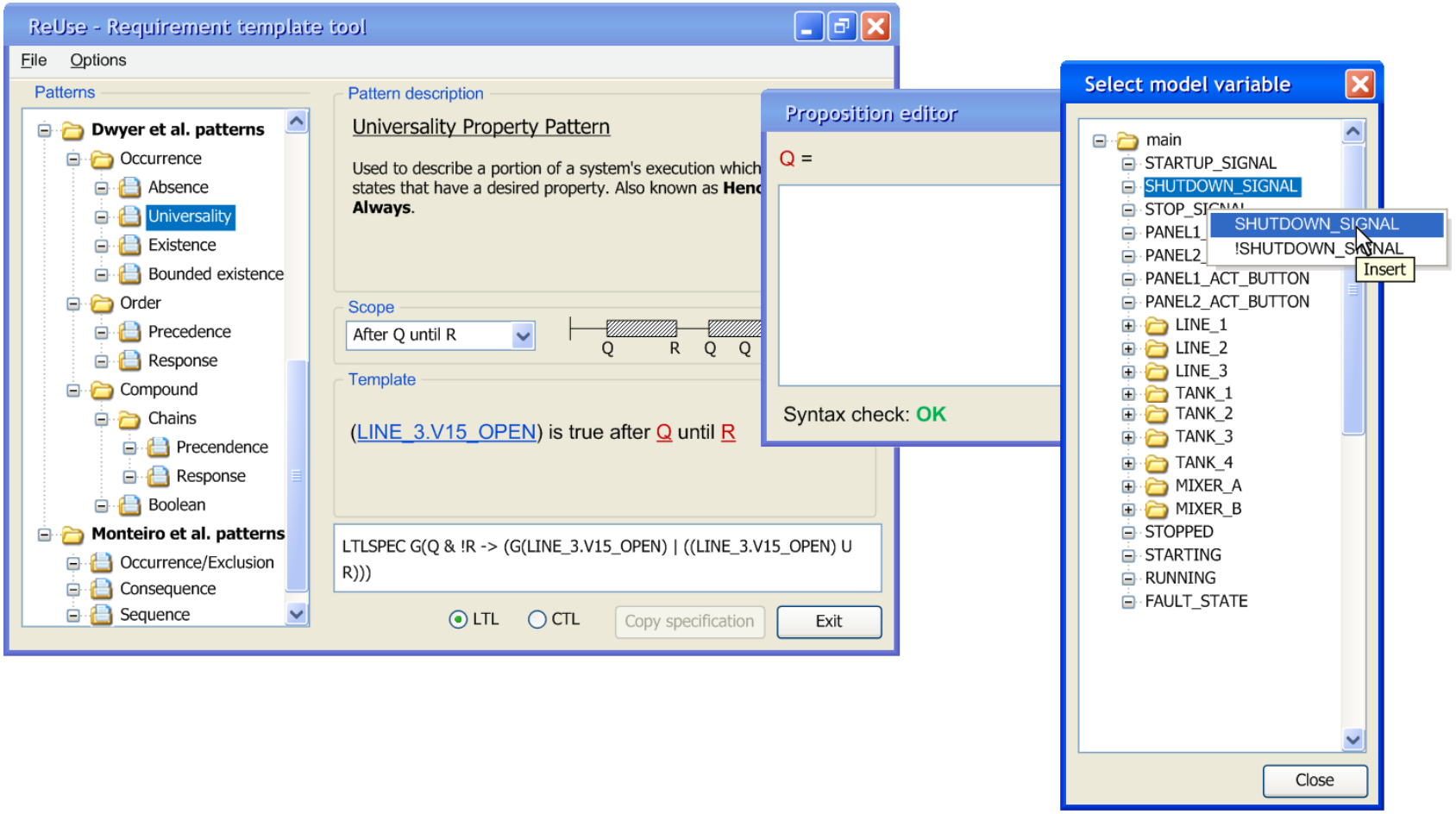
While the Reactor_controller_1 is in operational state Running and Temperature of Reactor_1 is above 100, the mode of Reactor_controller_1 is Auto.



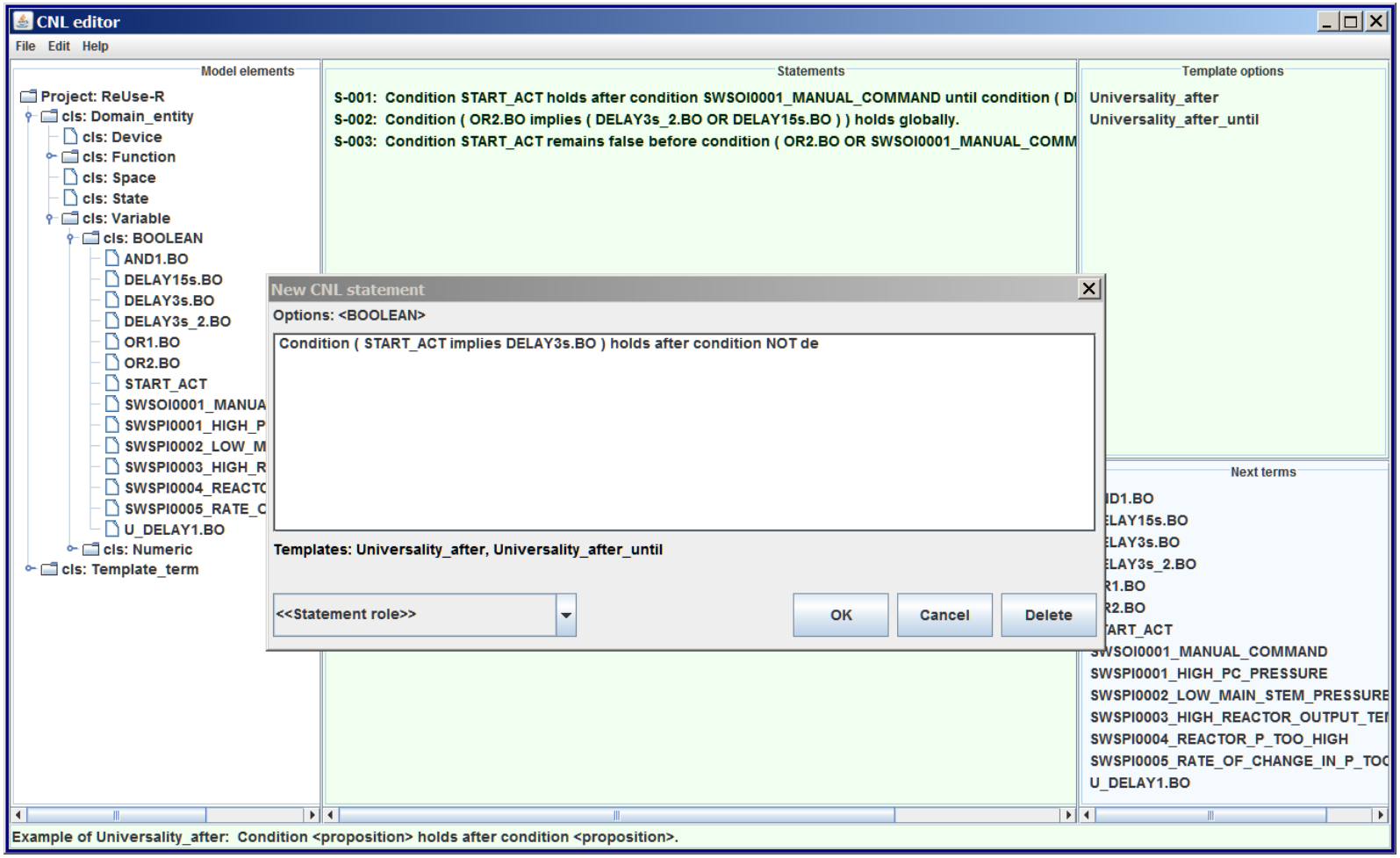
template Universality_global
sentence Condition <_proposition::P> holds globally.
formula LTLSPEC G(<P>)

LTLSPEC G ((Reactor1.Temperature > 100) & (Reactor_controller_1.state = Running)) -> (Reactor_controller_1.mode = Auto)

User interface option 1 – Fill in the blanks



Option 2 - Editing guided by templates with autocomplete



Statements and corresponding temporal logic formulas

Statement S-001
Sentence Condition START_ACT holds after condition SWSOI0001_MANUAL_COMMAND until condition (DELAY15s.clock is equal_to 0).

Statement S-002
Sentence Condition (OR2.BO implies (DELAY3s_2.BO OR DELAY15s.BO)) holds globally.

Statement S-003
Sentence Condition START_ACT remains false before condition (OR2.BO OR SWSOI0001_MANUAL_COMMAND).

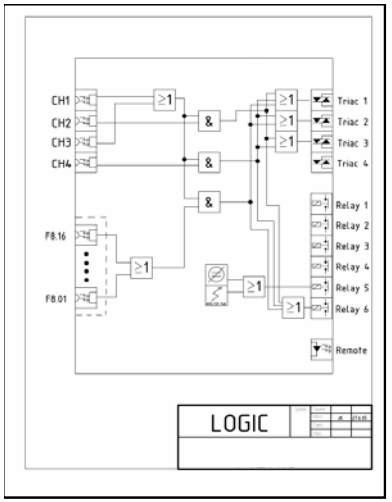
Statement S-004
Sentence Condition (START_ACT implies DELAY3s.BO) holds after condition NOT DELAY3s_2.BO until condition (DELAY15s.clock is equal_to 0).

```
LTLSPEC G(SWSOI0001_MANUAL_COMMAND & !(DELAY15s.clock = 0) -> (G(START_ACT) | (START_ACT U
(DELAY15s.clock = 0))))
LTLSPEC G((OR2.BO -> (DELAY3s_2.BO | DELAY15s.BO)))
LTLSPEC F(OR2.BO | SWSOI0001_MANUAL_COMMAND) -> (!START_ACT U (OR2.BO | SWSOI0001_MANUAL_COMMAND))
LTLSPEC G(!DELAY3s_2.BO & !(DELAY15s.clock = 0) -> (G((START_ACT -> DELAY3s.BO)) | ((START_ACT ->
DELAY3s.BO) U (DELAY15s.clock = 0))))
```

Conclusions / further research

- What kind of templates would be needed for real-world I&C systems?
- How should the CNL editor be implemented and integrated with other design / analysis tools?
- How should the user interface look like?
- Practical evaluation is needed!

- **http://www.vtt.fi/modelchecking/**



- STUK - Radiation and Nuclear Safety Authority, Finland
- Evaluation of NPP I&C system designs 2008-2011

- Fortum, Power company, Finland
- Verification of nuclear I&C by model checking 2012-