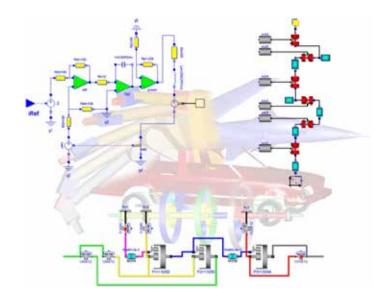
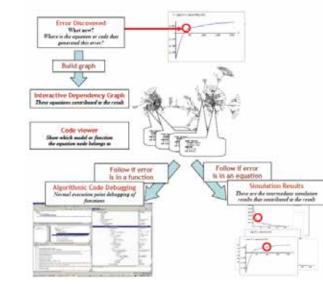
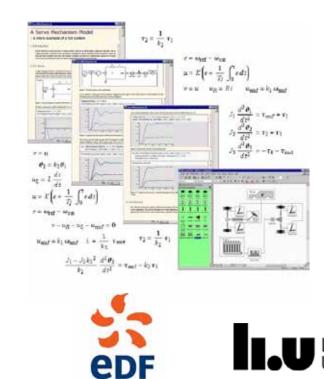
OpenModelica - The Common Requirement Modelling Language (CRML) Integration

Adrian Pop, Lena Buffoni, Audrey Jardin 2024-02-05 Open Source Modelica Consortium PELAB, Linköping University EDF, Électricité de France







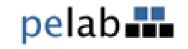
_INKÖPING

NIVERSITY





Modelica





What is CRML

- S The Common Requirement Modelling Language
- § CRML Tooling
 § The CRML Compiler

§ CRML Integration

- § OMEdit
- § VSCode
- § Online

Future work

- Solution The Common Requirement Modelling Language
 - **§** Language for Verifying Realistic Dynamic Requirements
- Started at **Started** at **Started** at **Started Started Started**
- § Further developed during the ITEA3

Ambition: Effective Engineering of Large CPS

Scope: Cyber-Physical Systems (CPS), especially energy systems
 Characteristics
 CPS Projects have often strong social and environmental impacts
 They are long lasting projects involving numerous stakeholders
 They should obey to multiple even conflicting requirements
 Project performance is a key as large over costs may be induced quickly due to financial charges (discount rate)

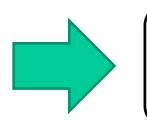
Challenges



- How to focus on conceiving systems more sustainable, trusty and resilient?
- How to solve over-constrained problems? How to coordinate stakeholders efficiently?
- How to specify the right need without going into realization details? How to reconcile innovation with what already exists?
- How to propagate changes in assumptions all over the system design cycle?
- How to evaluate design alternatives efficiently?
- How to perform failure modes, effects, and criticality analysis (FMECA) all along design lifecycle?
- How to justify and document design choices for future generations?

Examples of Challenges - Related to Energy Systems

- Interconnected systems with stringent physical constraints to ensure grid balancing
- **S** Long system lifecycles: new solutions built on existing ones (they are not created from scratch)
- S Compliance with strict safety and environmental rules
- S Compliance with dependability and availability constraints (to ensure security of energy supply)
- Involvement of multiple stakeholders: clients, regulatory authorities, grid operators, energy providers, insurers, urban and land-use planning, plant operators..., with different and possibly contradictory objectives
- Solution of Moving context with increasing uncertainties (due to geopolitical tensions, energy market instabilities, climate change, lack of energy policy coordination between countries, evolution of demand wrt. new usages...)

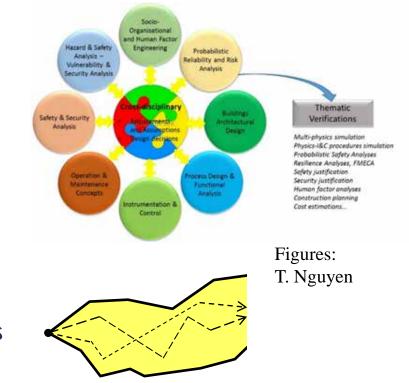


Energy systems are globally over constrained. New generation of methods & tools are needed to help engineers find the best compromise for covering multiple "what-if" operational situations (incl. variabilities and hazards)

S Today

- system evaluation is performed mostly with static models (or dynamics are considered too late)
- most verifications are performed manually (or with domain-specific tools) and hence not as often as necessary
- information is difficult to share between disciplinary engineering teams
- S There is a need for more rigorous engineering method to
 - Be more effective assessing the impact of each solution all along the system lifecycle including during preliminary design phases
 à guide and justify design choices also for non-experts
 - Open the solution space to innovative products or services
 a specify only "what is needed"

oversizing, late error detections, and eventually delays and cost overruns



CRML - A Part of the Solution

Idea =

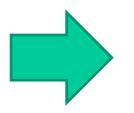
Use of **realistic dynamic behavioral models** to better handle multi-physics & systems' interactions **à e.g. Modelica**



Use of **formal dynamic requirement models** to automate verifications and evaluate multiple "what-if" scenarios **à CRML**

Rationale

- Consideration of "System Dynamics" as time may be part of new solutions to cover non-regular situations and hence source of cost reductions
- Formal verifications since for many CPS demonstration that the system operates safely is as important as the design itself



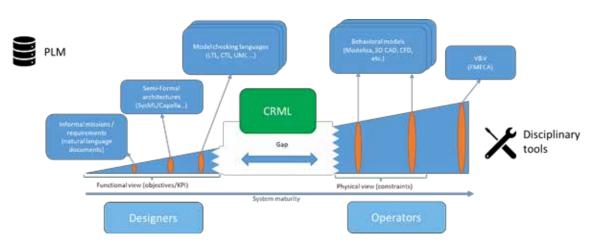
Scope of ITEA EMBrACE Project

"An enabler for making the best decisions at each step of the project cycle"

CRML: A Language for Verifying Realistic Dynamic Requirements

Why a new language?

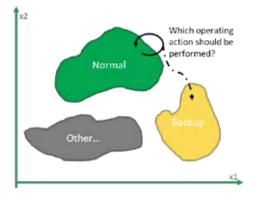
- Main principles from « System Engineering »
- Tools exists but are incomplete or essentially made for software design
- Native difficulty to address requirements that are
 « realistic » for systems with strong physical aspects
- In particular to study their dynamical interactions with their environments



CRML positioning vs. State-of-the-Art : a bridge between the physical & the functional views

A typical realistic dynamical requirement is multiple and stochastic

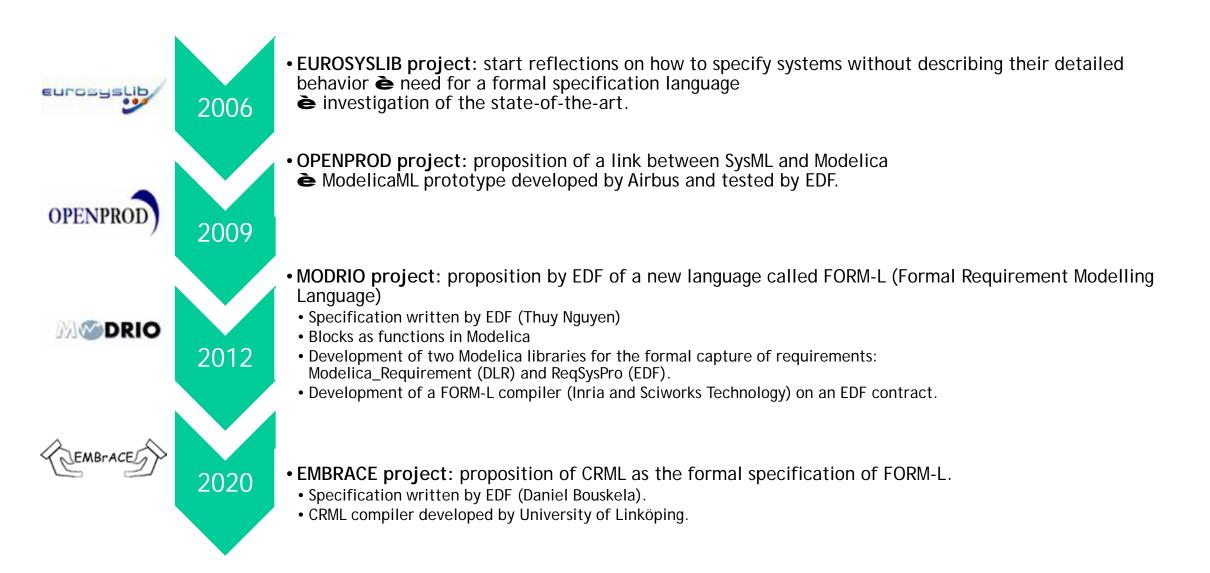
. . .



- 1. The system should stay within its normal operating domain.
- 2. If partial requirement 1 above fails, then the system should go back to its normal operating domain within a given time delay.
- 3. If partial requirement 2 above fails, or if partial requirement 1 fails with a too high failure rate, then the system should go to a safe backup state within a given time delay.
- 4. The complete requirement made of the conjunction of partial requirements 1, 2 and 3 should be satisfied with a given probability (e.g., > 99.99%).

... and a typical project quickly sees its complexity increase with the number of requirements/stakeholders and evolution over time

CRML: Not a Whim But a Long-Lasting History



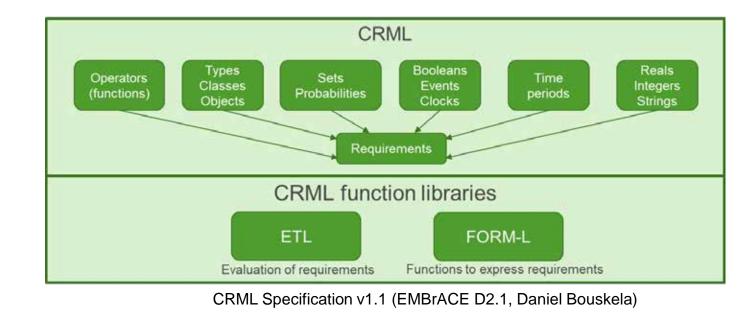
How To Express a CRML Requirement?

R = [Where or Which] [When] [What] + (optional) [How well]

for all' pump `in' system.pumps `during' system.inOperation `check count' (pump.isStarted `becomes true') `<=` 3; `during' systemOperatingLife `check at end' (estimator Probability (noStart at inOperation `becomes false')) `>' 0.99;

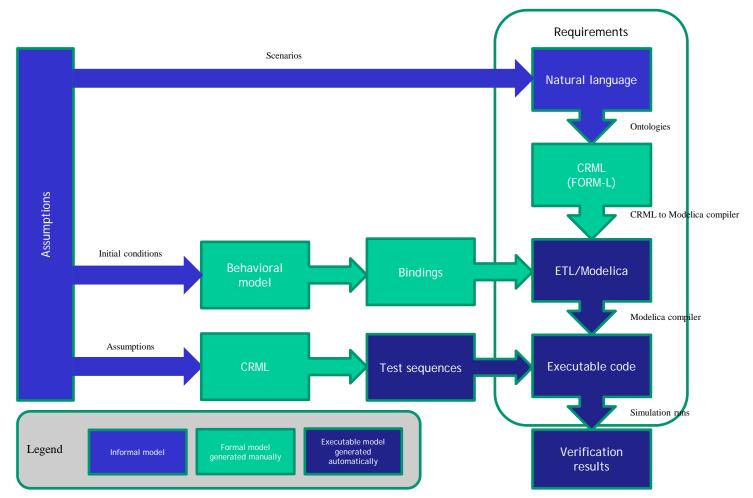
S Combination of 4 items

- Spatial locators
- § Time locators
- S Condition to be checked
- (optionally) Performance indicator
- S Value at instant t is a Boolean4 which can be : true, false, undefined Or undecided

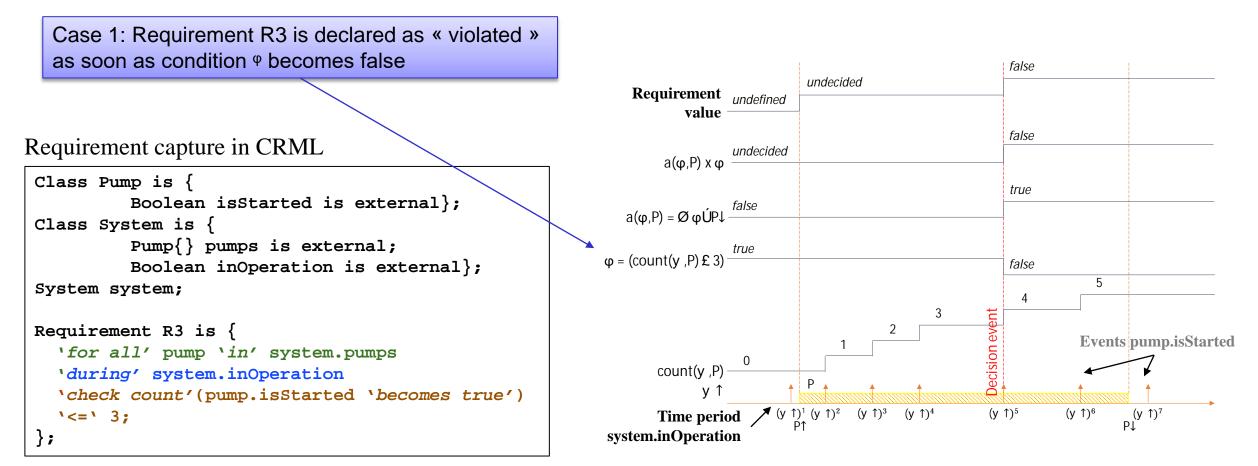


How to Use CRML for Verifications?

- Requirement models to capture all constraints on the system and define envelopes of acceptable behaviors
- Behavioral models to capture the behavior of design solutions
- Verification models to automate tests by using requirement models as observers to check whether design solutions meet requirements or not.



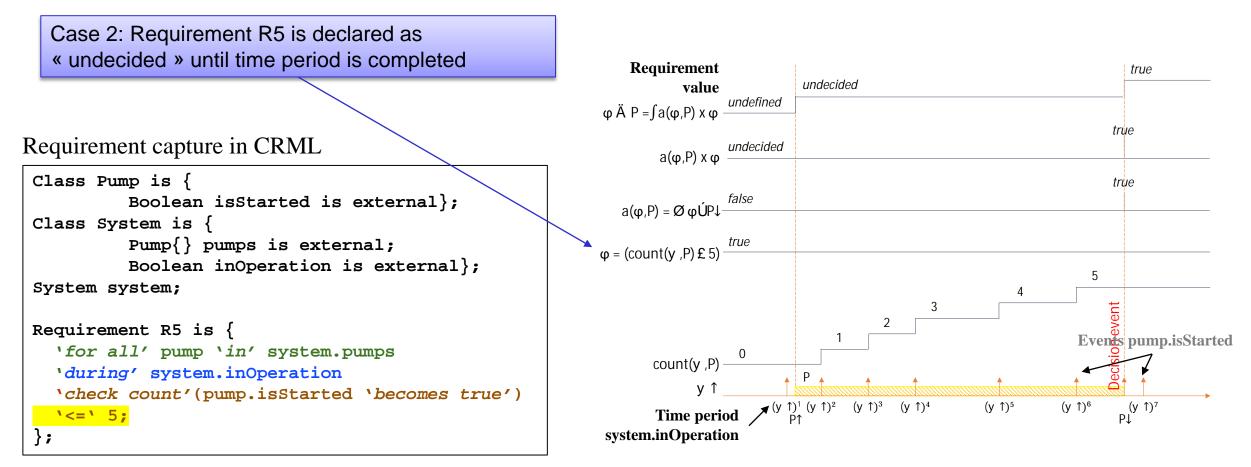
How To Evaluate a CRML Requirement?



external keyword is used to retrieve values in solution models Operators in " are defined by user to improve readability

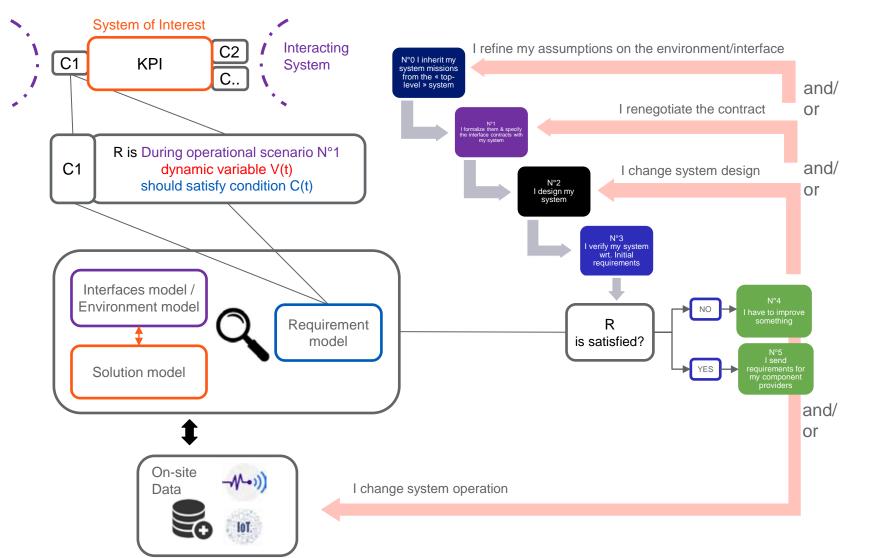
Requirement evaluation via observation of system behavioral dynamics

How To Evaluate a CRML Requirement?



Requirement evaluation via observation of system behavioral dynamics

How to Use CRML As a Decision Tool?



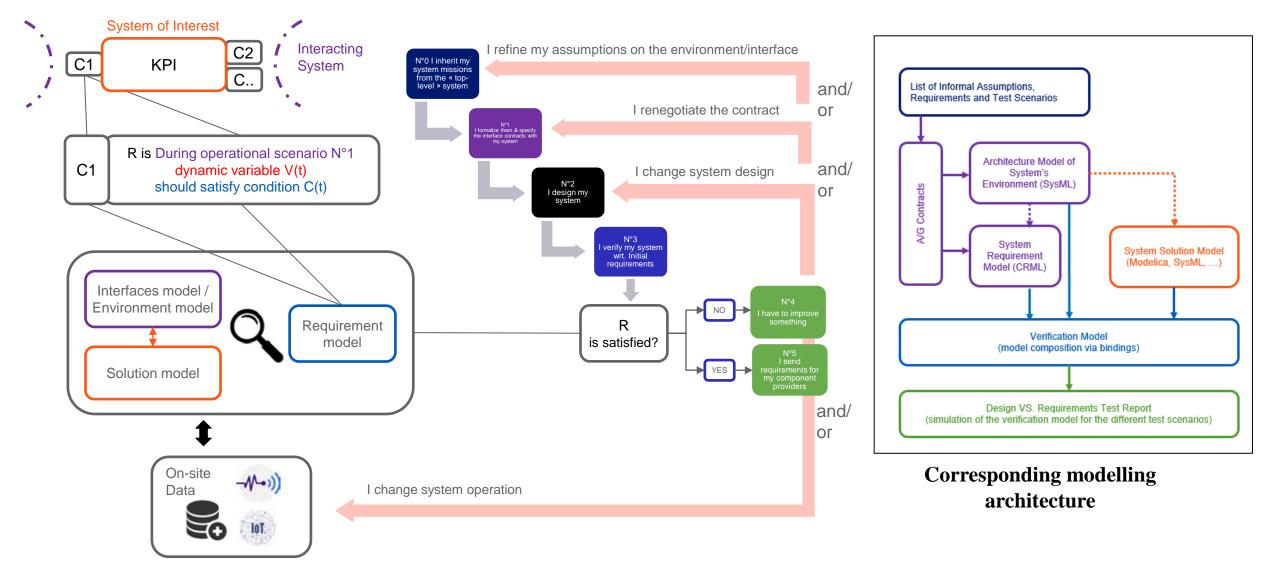
Model to support complexity

- **§** Scope of responsibility of stakeholders
- Multiplicity of constraints and operating scenarios
- Dynamics of interactions between systems, human and environment

Center development on the requirements

- Evaluate the impact of each solution on your overall ambition
- S Design only for the « right » need
- S Adapt the studies to « what is just needed »
- All along the project
- S And according to the data available at instant T

How to Use CRML As a Decision Tool?





§ What is CRML

§ The Common Requirement Modelling Language

S CRML Tooling S The CRML Compiler

§ CRML Integration

- § OMEdit
- § VSCode
- § Online

Future work



The CRML compiler

- <u>https://github.com/lenaRB/crml-compiler/</u>
- § Implemented in Java
- Translates CRML to Modelica
- Integrates with Unit testing and Reporting

Songoing work

- Support the full CRML specification
- Integrate with OpenModelica



Sew CRML menu in OMEdit

- Senerate and load Modelica code (also via the Library/File browser, right click)
- S Call the CRML compiler on the opened CRML file, generate Modelica code, load it into OMEdit, give errors if the code cannot be loaded

Ongoing

- § dialog for CRML configuration before compilation
- S annotation in the CRML file where one can provide a configuration which will be added as a Modelica annotation in the generated Modelica file

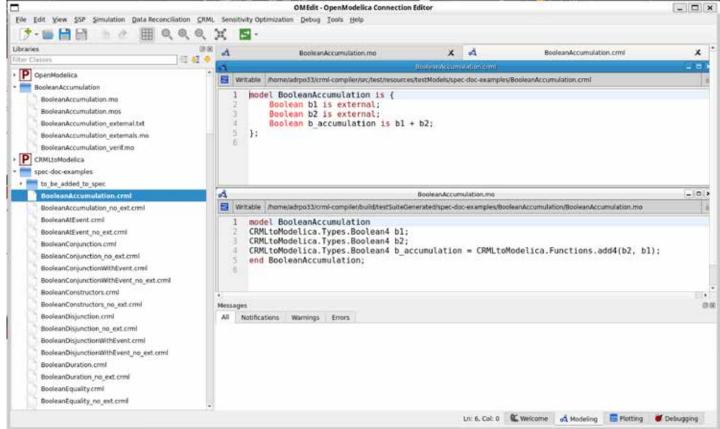
§ Run test report

- **§** Select a directory with CRML files
- S Call the CRML tool to generate the html report
- § Load and display the html report
- § A CRML test will go through these phases
 - § Parsing
 - Translation
 - Serification model generation
 - Execution
 - Sesult Verification

Sew / Open CRML models

🃑 Net	w CRML Model
들 Op	en CRML Model(s)
Op	en Directory

 § Load directories containing CRML models
 § Syntax Highlighting

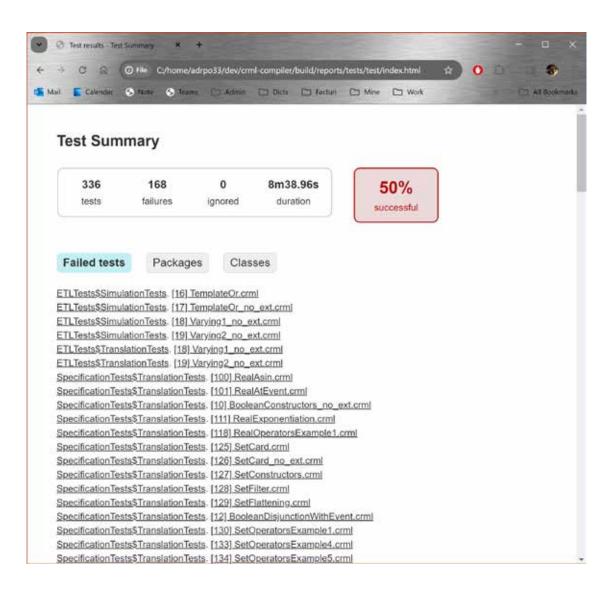


Generate and Simulate Modelica code

Eile Edit View SSP Simulation Data Reconciliation CR	OMEdit - OpenModelica Connection Editor ML Sensitivity Optimization Debug Tools Help	_ — X
👎 - 🔚 🔠 🔍 🔍 🗶 🛒 📢	• 🐔 🔀 🧶 🖉 🎎 🞬 🍾	
Libraries Ø 🕅	Piot : 1 X	Variables 🖉 🗷
Filter Classes 🛛 🖉 🛃 🕈		Filter Variables 🦉 🚱 🔶
P OpenModelica	🔀 💥 Grid 🚽 🗆 Log X 🔄 Log Y 🌿 💾 📥	Simulation Time Unit s *
- BooleanAccumulation	2 23 23 23	0-
BooleanAccumulation.mo	b_accumulation b2 b1	Time: 0 Speed: 1 -
BooleanAccumulation.mos	4	Speed.
BooleanAccumulation_external.txt		Variables Value Display Uni Description
BooleanAccumulation_externals.mo	1	- M (Activverif
BooleanAccumulation_verif.mo	3.5 -	✓ b1 4 ✓ b2 2
CRMLtoModelica		V b accumulation 4
spec-doc-examples	3	externals
- CRML		
• 🚯 UsersGuide	3 -	
TimeLocators		
Requirements		
+ 🙆 Blocks	2.5 -	
> 🗍 ETL		
Examples	· •	
🔸 💽 Tests	2	
• 272 Modelica	0 2 4 6 8 10 12 14	
Complex	time (s)	-
ModelicaServices	Messages 2018	0
M BooleanAccumulation	All Notifications Warnings Errors BooleanAccumulation_verif X	
CRML_test		
CRML_test.Spec_doc.BooleanAccumulation_externals	100% Cancel Simulation 🕵 Open Output File	t -
CRML_test.Spec_doc.BoeanAccumulation_verif	Compilation Output	
	The initialization finished successfully without homotopy method. * ### STATISTICS ### The simulation finished successfully.	
		- 1 • 1
	Ln: 3, Col: 11 🕊 We	elcome 💰 Modeling 🚍 Plotting 🖉 Debugging

Second CRML Testsuite

🕽 • 🖬 🗟 🖄 🗇 🖉	Data Reconciliatio	the second second second second second	OpenModelica Con Optimization Debu suite			(=)(-	X
braries Browser 🛛 🕲 🛃 📢	🗴 lmns.crml 🗴	a CategoryAssociati	ionExample1.crml 🗶	A CategoryAssociation.crm	X 🕺 Book	eanOperatorsExample1.crml	×
ber cursses 192 92 9	Writable	/home/adrpo33/crml	-compiler/src/test/res	ources/testModels/spec-doc-exam	nples/BooleanOp	eratorsExample1.crml	6
BooleanEquality.crml BooleanEquano_ext.crml BooleanFilter.crml BooleanFilter_no_ext.crml BooleanifThenElse.crml BooleanifThno_ext.crml Booleaninteno_ext.crml Booleaninteno_ext.crml BooleanNegation.crml	2 3 4 5 6 7 8 9 };	Boolean bl is Boolean b2 is Real y is sir Clock c is no Boolean b_out	at bl is equal s true; s true; n time; ew Clock (y > tl is (bl == t	to b2 when y become: 0);			
BooleanOpplo1.crml CategoryAssociation.crml CategoryAssample1.crml							
BooleanOpple1.crml CategoryAssociation.crml	Messages Bro	wser					810
BooleanOpplo1.crml CategoryAssociation.crml CategoryAssample1.crml CategoryConstructors.crml	Print and print and the second	wser cations Warnings	Errors				.89



CRML - OMEdit Integration

§ Editor Settings

bleanEquality.crml 1 bleanF 2	<pre>model BooleanOperatorsExample1 is { // Verify that b1 is equal to b2 when y becomes positive.</pre>	
blean	OMEdit - Options	[
oleani General	Colors	
bleani Seani Libraries bleani Text Editor bleani Modelica Editor bleani CRML Editor	Items: Text Number Keyword Type Quotes	Item Color:
olean 🗧 CompositeModel Editor	model HelloWorld is {	\
:egon Image: C/C++ Editor :egon Image: HTML Editor :egon Image: Graphical Views	};	
ckBoi Simulation ckCoi Messages		
ckCoi 1 Notifications ckCoi 2 Line Style ckCoi 2 Fill Style		
ckCu Plotting ckCu E Figaro		
ckDe CRML ckDis Debugger		
OMTLMSimulator OMSimulator/SSP Traceability		
* The changes will take effect	t after restart.	Reset Cancel

CRML - OMEdit Integration

Stool Settings

		OMEdit - Opti	MAR.V		
General	CRML				
Ubraries Text Editor	CRML Compiler Jar:	crml-compiler-all.jar			Browse
	CRML Compiler Arguments:	1			
Modelica Editor MetaModelica Editor CRML Editor CompositeModel Editor SSP Editor C/C++ Editor HTML Editor	CRML Processor:	java		Browse	Reset
CRML Editor	CRML Library Paths:			201	
CompositeModel Editor	/home/adrpo33/crml-compi	ler/resources/modelica_lib	aries		
SSP Editor		Add Path		Remove Path	
C/C++ Editor	/home/adrpo33/crml-compi	ler/resources/modelica_lib	aries		
HTML Editor		na an an tha na an			
🔏 Graphical Views					
Simulation					
Messages					
Notifications					
Line Style					
Channel of Contract of Contrac					
💓 Fill Style					
🐝 Fill Style 🔜 Plotting					
Plotting					
Plotting					
Plotting Figaro					
Plotting Figaro CRML Debugger					
Plotting Figaro CRML Debugger MI					
Plotting Figaro CRML Debugger FMI					
Plotting Figaro CRML CRML Debugger FMI OMTLMSimulator OMSimulator/SSP Traceability					
Plotting Figaro CRML Debugger FMI OMTLMSimulator OMSimulator/SSP					
Plotting Figaro CRML CRML Debugger FMI OMTLMSimulator OMSimulator/SSP Traceability					

Sasic VSCode extension

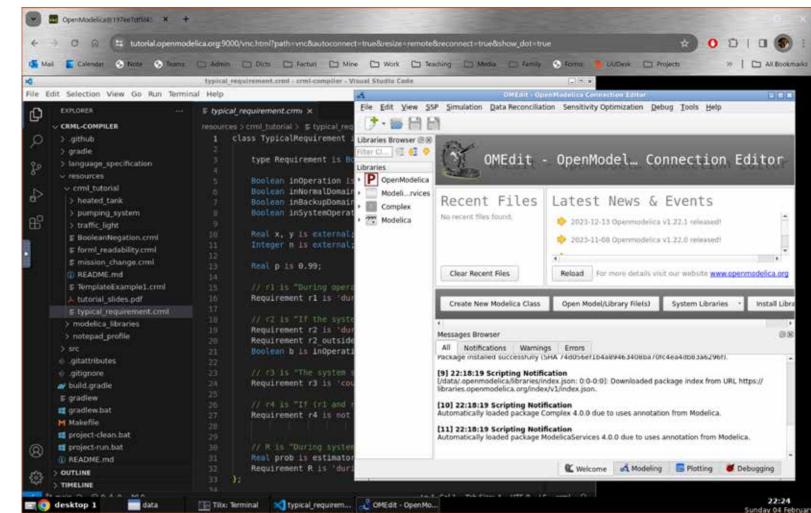
- <u>https://github.com/lenaRB/crml-vscode</u>
- syntax highlighting

×1 =	ile Edit Selection View Go Run \cdots \leftarrow	→ O cml-compiler 00 -	x
Q	EXPLORER ····	E BooleanAccumulation_no_ext.crml ×	œ۰
2 🤉 🤹 🖓 🕼 🗄 🗏 🖓	V OPEN EDITORS V OPEN EDITORS V Societan Accumulation_no_ext.crml incitest/veso Boolean Accumulation_no_ext.crml incitest/veso CRML-COMPILER V Soc V test V resources V testModels bibraries v spec-doc-examples bibraries v spec-doc-examples bibraries v spec-doc-examples bibraries bibraries	<pre>src > test > resources > testModels > spec-doc-examples > E BooleanAccumulation_no_extcrml model BooleanAccumulation_no_ext is { Boolean b_true is true; Boolean b_false is false; Boolean b_undetided is undefined; //Test truth table for the Boolean accumulation operator //First rom Boolean b_true_accumulates_true is b_true + b_true; //value is true Boolean b_true_accumulates_true is b_true + b_undefined; //value is true Boolean b_true_accumulates_true is b_true + b_undefined; //value is true Boolean b_true_accumulates_true is b_true + b_undefined; //value is true Boolean b_true_accumulates_true is b_true + b_undefined; //value is true Boolean b_true_accumulates_true is b_false + b_true; //value is false Boolean b_talse_accumulates_true is b_false + b_true; //value is false Boolean b_false_accumulates_true is b_false + b_true; //value is false Boolean b_false_accumulates_true is b_false + b_undefined; //value is false Boolean b_false_accumulates_true is b_false + b_undefined; //value is false Boolean b_false_accumulates_true is b_false + b_undefined; //value is false Boolean b_false_accumulates_true is b_false + b_undefined; //value is false Boolean b_false_accumulates_true is b_undecided + b_true; //value is false Boolean b_mudecided_accumulates_true is b_undecided + b_false; //value is false Boolean b_undecided_accumulates_true is b_undecided + b_false; //value is false Boolean b_undecided_accumulates_true is b_undecided + b_false; //value is false Boolean b_undecided_accumulates_undefined is b_undecided + b_fulse; //value is false Boolean b_undecided_accumulates_undefined is b_undecided + b_undefined; //value is undecided foolean b_undecided_accumulates_undefined is b_undecided + b_undefined; //value is false Boolean b_undecided_accumulates_undefined is b_undecided + b_undefined; //value is undecided foolean b_undecided_accumulates_undefined is b_undecided + b_undefined; //value is undecided //f</pre>	
§ (2)	F BooleanDisjunction.crml	PROBLEMS CONTOLL COMPOLE CONSOLL TERMINAL PORTS PS C:\home\adrpo33\dev\crml-compiler>	8 ^ 3
605	> JAVA PROJECTS		

CRML - Online

SCRML and OpenModelica tutorial available online

- <u>https://tutorial.opennmodelica.org/</u>
- § No install needed





§ What is CRML

- **§** The Common Requirement Modelling Language
- § CRML Tooling
 § The CRML Compiler

§ CRML Integration

- § OMEdit
- § VSCode
- § Online

§ Future work

Sear Future

- Selease the OMEdit integration as part of OpenModelica (v1.23.0)
- Set more feedback from EDF on the prototype via evaluation on industrial use-cases

§ Future

- § How to group together several requirements into a project
- Section 10 Section
- Sevaluate traceability from CRML to simulation results
- Integration with dashboards to aggregate requirement information



Thank You! Questions?

The CRML Project https://crml-standard.org/

The OpenModelica Project https://www.OpenModelica.org