

Generate Code from **UML/SYSML** for Improved Traceability

Generate Misra-compliant C/C++ code to enhance traceability from your requirements to your source code using Embedded Engineer 3.0



EMBEDDED ENGINEER[©]

Low Code Platform for
Embedded Systems



by LieberLieber Software

<https://www.lieberlieber.com/embedded-engineer/>

<https://help.lieberlieber.com/display/EmbeddedEngineer/Embedded+Engineer>

Main Questions/Doubts

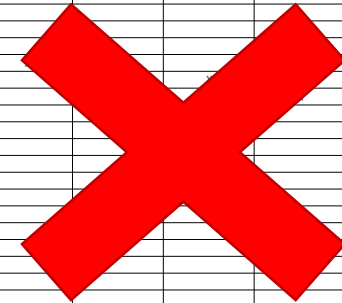
- Why have you already “won” if you only generate .h files?
- Where are the advantages of code generation in general?
- Do I always have to model everything?
- How can I trace requirements down to the source code and use the great advantages of traceability?
- Customizations or Adaptations? How can I modify the generated code?
- How about Round Trip Engineering?



How to ensure consistency?

Traceability-Tables?

	Requirement 1	Requirement 2	Requirement 3	Requirement 4	Requirement 5	Requirement 6	Requirement 7	Requirement 8	Requirement 9	Requirement 10	Requirement 11	Requirement 12	Requirement 13	Requirement 14	Requirement 15	Requirement 16	Requirement 17	Requirement 18	Requirement 19
Test T1		X																	
Test T2		X		X															
Test T3				X															
Test T4					X														
Test T5																			
Test T6																			
Test T7																			
Test T8			X		X														
Test T9																			
Test T10																			
Test T11	X																		
Test T12																			
Test T13																		X	
Test T14																			
Test T15	X																		
Test T16			X																
Test T17																			
Test T18						X								X					
Test T19									X		X								
Test T20										X									
Test T21											X								
Test T22																			
Test T23																			
Test T24																			
Test T25																			
Test T26																			
Test T27																			
Test T28																			X
Test T29																			
Test T30																			
Test T31																			
Test T32																			
Test T33																			
Test T34						X													
Test T35				X															
Test T36																			
Test T37		X																	
Test T38	v																		



Create Code from Models

```
class InterfaceTest
{
public:

    /// auto generated virtual destructo
    virtual ~InterfaceTest() {}

    /// This is the description of metho
    /// @param param1: this is the descr
    ///
    /// @covers REQ-1
    virtual bool doA(uint8 param1) = 0;

    /// this is the description of metho
    virtual void doC() = 0;

};
```

codeBeamer

My Start Projects Reports Wiki Documents Tracker

Power Window Controller > Trackers

System Requirement Specifications > All Items

Type to filter

System Requirement Specifications

Functional Requirements

construct MBD_UML_Req_002 - Construction View

Eco-Friendliness
tags
cb_issue_id = /item/1665
cb_url = http://dsiegl-nb.olymp3000.com:8080/db/
last_import = 26.06.2019 13:46:46
revision = 1

(from FromCodeBeamer)

Maximum time in moti

The travel time from fully open to commanding motion continuously!

Auto lower

The auto-lower feature is triggered by the controller will continue to comma

DEMO Slot 1

Create code from Interfaces and Classes

Generate Code from behaviour Models

Lot's of new and old challenges ahead!

- Functional Safety (ISO 26262,...)
- UI complexity
- Traceability
- Documentation requirements



DEMO Slot 2

Create code from Behavior

Roundtrip?

- Reverse for Libraries and Legacy Code
- Code Sync for operation bodies
- FULL Roundtrip not useable in the real world



DEMO Slot 3

Reverse/Code Sync

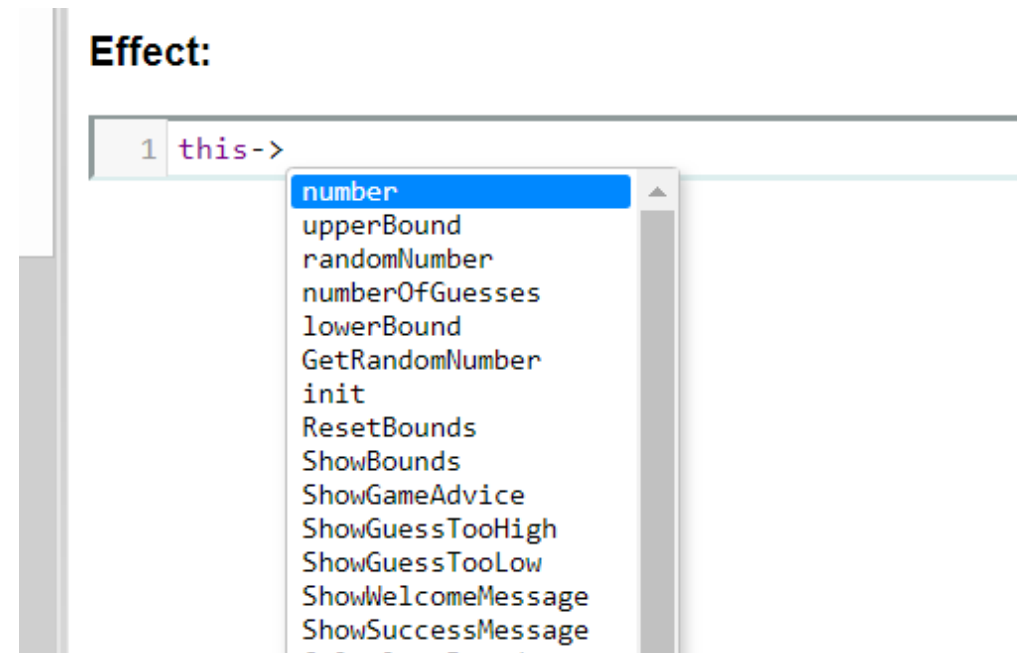
Code from UML/SysML

Generation

- Interfaces
- Classes/Blocks
- Activity Diagrams
- State Machines

Full behavior can be modeled with Activities but Code is sometimes more efficient!

NEW Code Editor Bridges the Gap



Key Take Aways

- Just .h files with traceability info can be a first step
- Expanded traceability
- Integrate Legacy Code and Libraries
- Link Requirements with Code in the Model
- Easy to modify the layout of the Code
- The truth on Round Trip Engineering



Call to action

- Getting Started
 - <https://help.lieberlieber.com/>
- Download Embedded Engineer 3.0
 - <https://www.lieberlieber.com/embedded-engineer/>
- Arrange a WebMeeting
 - welcome@lieberlieber.com



Thank You! Questions?

Richard Deininger

Daniel Siegl

