

# User Perceptions of Code and Model Generation with Simulink in Industrial Settings – A Survey

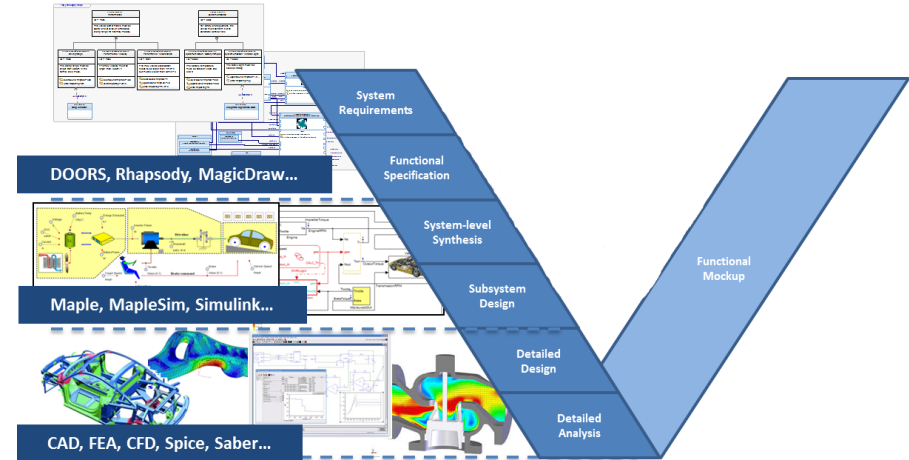
**Johan Cederbladh**, Jakob Norin

# MBD in industrial settings

Model-Based Development (MBD) has a wide range of tools and application areas.

There are many capabilities like code generation which make MBD attractive.

In some industrial segments the majority if not all industrial companies use MBD tools.

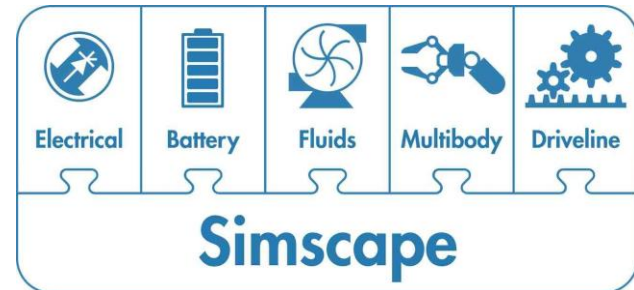
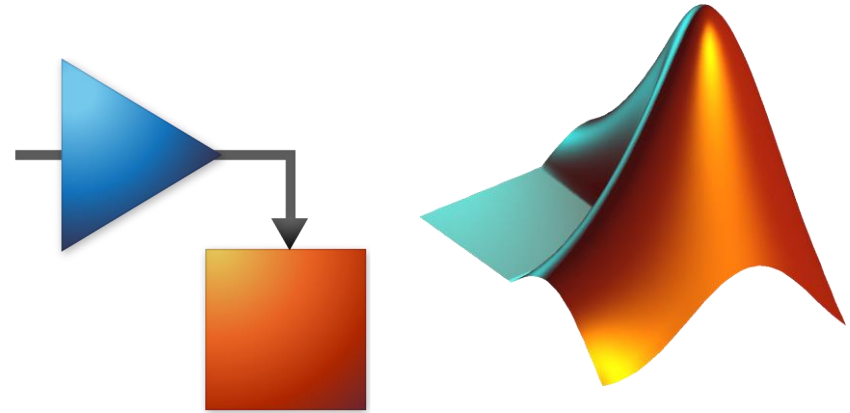


# MathWorks

MathWorks offers a popular MBD tool-chain where many students are trained in the tools.

Simulink is attached to the MathWorks license and is often coupled with other libraries and tools from this license.

Simulink enables code-generation through standard libraries and offers modularity in architecture models.

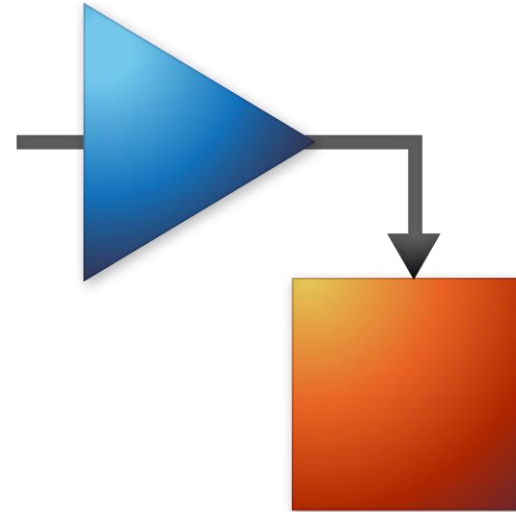


# Our context

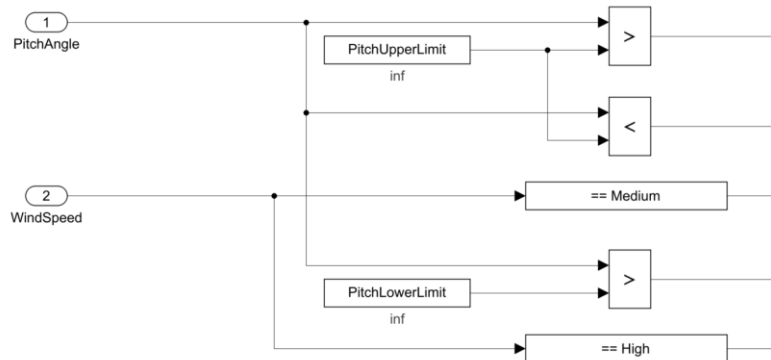
We consider an automotive OEM context.

During development, Simulink is used to generate test code for hardware platforms.

The use of Simulink has been integrated in the context for a number of years and is part of the best practice.



# Our context



```

/*
 * File: untitled.c
 *
 * Code generated for Simulink model 'untitled'.
 *
 * Model version           : 1.0
 * Simulink Coder version   : 9.6 (R2021b) 14-May-2021
 * C/C++ source code generated on : Fri Sep 26 09:38:15 2025
 *
 * Target selection: ert.tlc
 * Embedded hardware selection: Intel->x86-64 (Windows64)
 * Code generation objectives: Unspecified
 * Validation result: Not run
 */

#include "untitled.h"
#include "untitled_private.h"

/* External inputs (root inport signals with default storage) */
ExtU_untitled_T untitled_U;

/* External outputs (root outports fed by signals with default storage) */
ExtY_untitled_T untitled_Y;

/* Real-time model */
static RT_MODEL_untitled_T untitled_M;
RT_MODEL_untitled_T *const untitled_M = &untitled_M;

/* Model step function */
void TA_untitled(TA_untitledArgs_t* args)
{

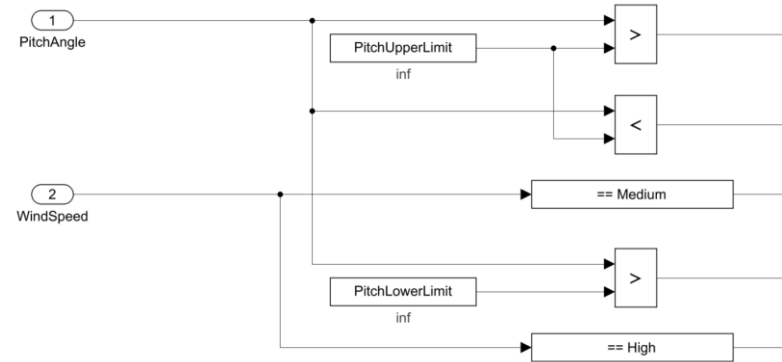
```

# Some general observations

In the industrial context we note that Simulink is a divisive tool, also matching with general findings in research.

Particularly, the use of MBD tools like Simulink are highly appreciated but also controversial due to the same aspects of graphical coding.

We would like to understand the reasons for the nuances in Simulink's reception. Particularly, in regard to aspects related to collaboration and usability.



# A survey

We thought a survey could be interesting to understand the concept better.

We launched a survey at the industrial context, and received 23 responses from practitioners.

Likert scale 1 – 5, where 1 == poor and 5 == excellent

We ask about the following topics;

1. Respondent information.
2. User friendliness.
3. Performance.
4. Quality of generated artifacts.
5. Collaboration.
6. Scalability.
7. Other.

# Respondent information

We receive 23 responses.

Relatively heterogenous audience in both experience and roles.

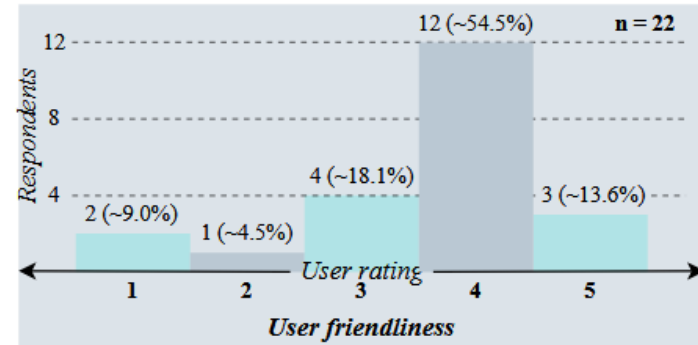
Somewhat varied use of Simulink across roles.

<b>Years of Experience</b>	<b>0–1</b>	<b>2–4</b>	<b>5–10</b>	<b>10+</b>
Hardware Engineer	1	1	1	–
Software Engineer	1	6	6	2
Test Engineer	–	2	1	1
System Architect	–	1	2	–
Researcher	1	2	2	–
Other	1	2	1	1

# User friendliness

We see that most users find the user friendliness high in Simulink.

The lower responses are mostly from engineers in software engineering.

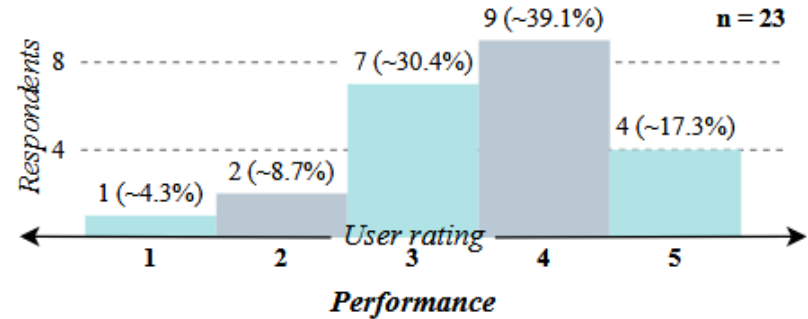


# Performance

Performance has one the highest number of 5 scores, but also a large number of responses in the middle of the road.

Practitioners who care about performance seem to rate Simulink highly, while many also do not have a strong opinion about performance in Simulink.

Performance might be interpreted differently across usage of the tool.



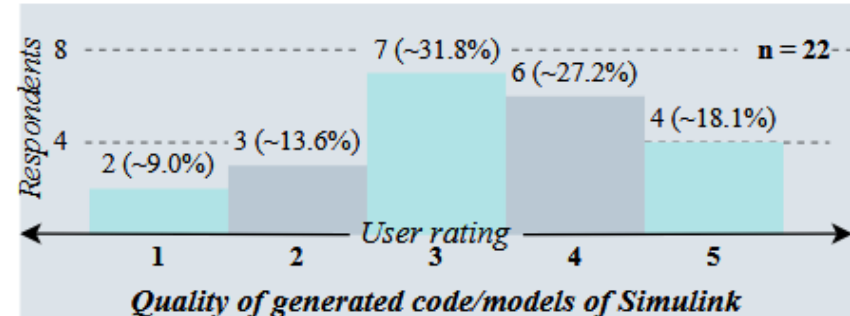
# Quality of generated code/models

The quality of generated code/models has a wide range of responses.

We can see that much of these experiences connect to the role of respondents.

We also note that some respondents do not interact with generated code to a large extent.

The domain of the user plays a role in their evaluation of the generated artifacts.

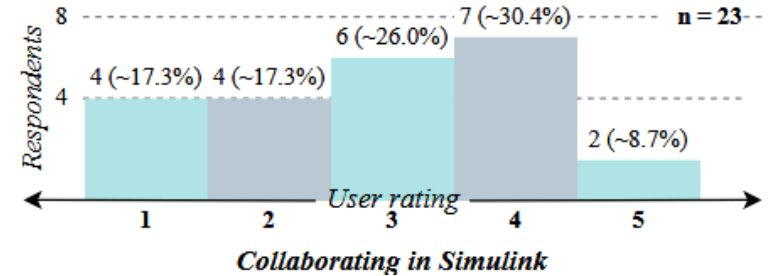


# Collaboration

Collaboration in Simulink has the lowest average score among the questions, with the most 1 scores of any question.

While some questions have a specific domain that puts lower scores, in collaborations it is shared between most.

Collaboration challenges relate to multiple users, lack of Git, sharing of models, etc.

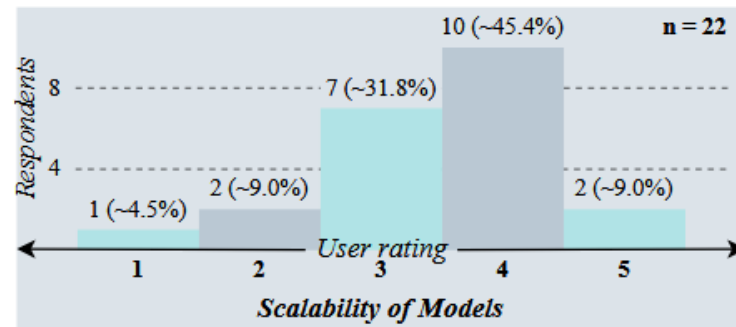


# Scalability

Scalability is similar to performance as many answers are in the middle of the road, with few practitioners having strong opinions.

One aspect might be that many practitioners do not have scalability as a main concern during their work.

Since the use of Simulink is mature at the company this might also have led to more mature scalability in workflows.

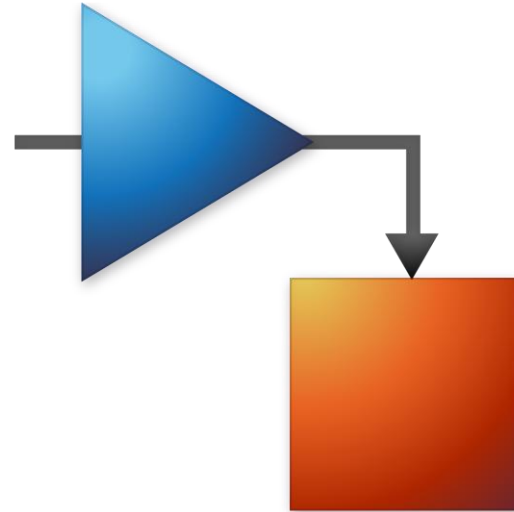


# Results summary

We see a mostly positive reception of Simulink.

The most negative area is that of collaboration, where free-text answers provide a number of critiques.

We can see that software engineering practitioners have significantly lower positive rates of responses compared to other domains.



# Discussion and outlook

MBD is probably here to stay and does seem to have a positive reception.

Software engineering practices seem to be challenging to support in MBD and causes more negative perceptions among software engineers.

The survey was relatively simplistic.

The findings seem to align with similar studies.

Since we specifically observe Simulink there is a reduced generalizability towards MBD as a whole, which needs to be considered.

# Thanks!

[johan.cederbladh@mdu.se](mailto:johan.cederbladh@mdu.se)