LLM ASSISTED NO-CODE HMI DEVELOPMENT FOR SAFETY-CRITICAL SYSTEMS
Insights of a Short Impirical Study

Prof. Dr. Matthias Harter
November 2023
A (VERY) SHORT RÉSUMÉ

Contact information at the end of this presentation...
SHORT RÉSUMÉ

Some call it CV...

Name  Prof. Dr. Matthias Harter
Fields of interest / profession
  • Patents and IP
  • AI, AGI and humanity
  • ASICs, Circuits and Systems
  • Aviation, Simulators

since 07/11  Professor for Embedded Systems and Microcomputers
             Hochschule RheinMain
             University of Applied Sciences

10/12 – 09/18  Head of the Department of Electrical Engineering and Information Technology

10/17 – 10/23  Head (founder) of new study program „Electrical and Aviation Engineering“
STATE OF THE ART: EMBEDDED SOFTWARE FOR SAFETY-CRITICAL APPLICATIONS

Examples from Aviation Engineering
A320 COCKPIT
Safety-Critical Embedded SW development today
CESSNA 172 (4 SEATS)
Replacement of analog instruments
Scenario (example): Artificial horizon used as Primary Flight Display (PFD) in an aircraft's cockpit.

State of the art: Automation of the development process with certified tools, e.g., Ansys SCADE
- Visual part specified as graphical models based on OpenGL primitives
- Usage of pre-defined widgets (fast!)
- Functionality as SysML models
- SW code is generated from models, (almost) no hand-written source code

=> Safety-critical! Can AI assist? How?
ANSYS SCADE
Certified tool, used by Airbus, Boeing, etc.
for Model Based Systems Engineering (MBSE)
“IN THE BEGINNING WAS THE WORD”
Requirements written in natural language are perfect for LLMs (e.g., GPT-4)
SCENARIO FOR THE FUTURE

How realistic (how wise) is it to use AI in safety-critical applications?
FUTURE:
Graphical and functional model, with interface to Model Based Systems Engineering (MBSE) tool

Step 1
Requirements written in natural language are fed into LLM (e.g., GPT-4).

Step 2
LLM generates API calls for the development tool to create and connect the instances of the **graphical and functional model** of the embedded SW.

Models are transferred into internal representation by development tool.

Models are analyzed and edited by human engineer in the development tool, only **when necessary** (human-in-the-loop policy).

C/Ada source code is generated automatically by code generator (e.g., KCG).

Source code is compiled into binary (executable) for adaption by Real-Time-OS (RTOS)

Alternative: LLM generates models directly in native file format (e.g., XML) of MBSE tool

*12.11.2023*
V-MODEL OF DEVELOPMENT PROCESS
Crucial: Usage of LLM (AI) in which step?

System concept
discussion, decision-making, etc.

S/W requirements
e.g. IBM DOORS, MS Office, etc.

S/W design
e.g. Ansys SCADE, Matlab/Simulink, IBM Rhapsody, etc.

S/W coding
e.g. SCADE KCG, Matlab Coder, etc.

RTOS Adaption
building executable(s)

System requirements allocated to S/W
High-level requirements
Graphical and/or functional model
Source code

Increasing degree of automation

Decreasing alignment effort
METHODOLOGY FOR EVALUATION

Evaluation of the capabilities of current LLMs for a limited test case
TODAY: LIMITATIONS
Limited to graphical model, without direct (automatic) interface to development tool editor

Requirements written in natural language are fed into LLM (e.g., GPT-4).

LLM generates code (e.g., Python, TikZ/LaTeX) for the graphical models of the embedded SW only.

Code is executed by interpreter and graphical models displayed.

Graphical models are analyzed and transferred into development tool editor manually by human engineer.

C/Ada source code is generated automatically by code generator (e.g., KCG).

Source code is compiled into binary (executable) for adaption by Real-Time-OS (RTOS).
1. General Layout & Dimensions:
   - The PFD shall have a rectangular aspect ratio suitable for installation in standard cockpit instrument panels.
   - The sky and earth shall be perfectly aligned at the horizon line.
   - The horizon line shall be centered horizontally on the PFD, and its vertical placement shall adjust based on the aircraft’s pitch angle.

2. Color and Appearance:
   - The PFD shall represent the sky in blue.
   - The PFD shall represent the earth in brown.
   - The horizon line shall be a distinct, bold white line for easy visibility against both the sky and earth backdrops.
RESULTS: REQUIREMENTS 1...9 AND GPT-4
100% fulfillment achieved

3. ... 7. omitted for clarity
8. Additional Flight Information:
   • The PFD shall display other pertinent flight data such as vertical speed, angle of attack, and barometric pressure.
   • This information should be arranged in a manner that does not clutter the primary attitude information.
9. Warning and Caution Indicators:
   • The PFD shall have provisions for displaying warning (red) and caution (amber) indications for critical flight parameters, such as stall warnings or autopilot disengagement.
Requirements not met:

6. **Heading Indicator:**
   - ...
   - The current heading shall be indicated by a **fixed pointer or triangle**, with the tape/rose rotating behind it.

7. **Turn Coordinator:**
   - The PFD shall incorporate a turn coordinator, represented by a curved line or other suitable graphical representation, to show the **rate and direction of turn**.

8. **Additional Flight Information:**
   - The PFD shall display other pertinent flight data such as **vertical speed**, **angle of attack**, and **barometric pressure**.
RESULTS FOR ALL 9 REQUIREMENTS
Only GPT-4 and GPT-3.5 succeeded

- GPT-4 always generated error-free code
- GPT-3.5 generated correctable code
- Other LLMs tested not ready for integration

Not to be confused:
Code generated by LLM vs.
Code generated by KCG

<table>
<thead>
<tr>
<th>LLM</th>
<th>Degree of fulfillment</th>
<th># of error-free code variantes</th>
<th># of correctable code variants</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT-4</td>
<td>Min. 37% Median 74% Max. 100%</td>
<td>14 of 14</td>
<td>N/A</td>
</tr>
<tr>
<td>GPT-3.5</td>
<td>Min. 16% Median 39% Max. 68%</td>
<td>7 of 8</td>
<td>1 of 8&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>CodeLlama</td>
<td>0%</td>
<td>0 of 2&lt;sup&gt;b&lt;/sup&gt;</td>
<td>0 of 2&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td>StarChat</td>
<td>0%</td>
<td>0 of 2&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0 of 2&lt;sup&gt;c&lt;/sup&gt;</td>
</tr>
<tr>
<td>CodeGen2.5</td>
<td>0%</td>
<td>0 of 2&lt;sup&gt;c&lt;/sup&gt;</td>
<td>0 of 2&lt;sup&gt;c&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

Footnote:
<sup>a</sup> contained errors that GPT-3.5 corrected after being instructed
<sup>b</sup> code output ended after approx. 5000 characters
<sup>c</sup> timeout after several minutes without any output
RESULTS FOR REQUIREMENTS 1 TO 5
CodeLlama now produced code

- CodeLlama generated model for shorted list of requirements due to restricted context window
- Other LLMs tested still not usable

<table>
<thead>
<tr>
<th>LLM</th>
<th>Degree of fulfillment</th>
<th># of error-free code variants</th>
<th># of correctable code variants</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPT-4</td>
<td>100%</td>
<td>8 of 8</td>
<td>N/A</td>
</tr>
<tr>
<td>GPT-3.5</td>
<td>Min. 64% Median 85% Max. 96%</td>
<td>4 of 4</td>
<td>N/A</td>
</tr>
<tr>
<td>CodeLlama</td>
<td>Min. 14% Median 29% Max. 86%</td>
<td>0 of 6</td>
<td>6 of 6&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>StarChat</td>
<td>0%</td>
<td>0 of 2&lt;sup&gt;b&lt;/sup&gt;</td>
<td>0 of 2&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td>CodeGen2.5</td>
<td>0%</td>
<td>0 of 2&lt;sup&gt;b&lt;/sup&gt;</td>
<td>0 of 2&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
</tbody>
</table>

Footnote:
<sup>a</sup> repeatedly the same error using Qt (code could be corrected manually)
<sup>b</sup> timeout after several minutes without any output
CODELLAMA VARIANTS

Best results (57% and 86% fulfillment) for shorted list of requirements (requirements 1...5)
CONCLUSION

LLMs / AI can assist human engineers, but should never replace them completely
SCENARIO
AI acting as assistant to engineering teams

Challenge / Risk:
What if the assistant (AI) becomes more experienced, efficient and reliable than the human team?
⇒ get rid of human-in-the-loop policy?
⇒ humans only for high-level requirements?
REFERENCES AND CONTACT INFORMATION

Comments and discussion always welcome!
SELECTED REFERENCES

Full list: see paper


SELECTED REFERENCES

Full list: see paper


THANK YOU FOR LISTENING

Contact
Prof. Dr. Matthias Harter
Faculty of Engineering
Department of Electrical Engineering and Information Technology
Am Brückweg 26
D-65428 Rüsselsheim
+49 6142 898-4223
matthias.harter@hs-rm.de