Outline

Introduction

Context

Contribution overview

Application to System Engineering

Evaluation

Demo

Conclusion
Introduction

LLM and System Engineering

- AI is now used for many applications
- Could LLM (e.g., ChatGPT) be used to assist system architects?
  - Modeling, verification, coding

Outline

- Context: methodologies for designing embedded systems, ChatGPT (and similar AI)
- Contribution on S.E.: general approach, implementation aspects, evaluation
- Demo, conclusion
Outline

Introduction

Context

Contribution overview

Application to System Engineering

Evaluation

Demo

Conclusion
ChatGPT in a Nutshell

Training

embeddings

Reinforcement Training

Query

prompt

Text tokenization

Text generation

Answer

Our Idea: AI&ChatGPT to Support System Engineering

(Model-Based) System Engineering = document understanding and transformation into views/diagrams/code

**Input**
- Improve ChatGPT to understand inputs such as:
  - System specification
  - Objective of the request: identifying requirement, identifying system blocks, etc.
  - Constraints from system architects
  - Existing diagrams or code

**Output**
- Drive ChatGPT to produce answers related to system engineering
- Drive ChatGPT to produce an easy-to-parse output
- Automatically identify if produced outputs are invalid/wrong, and force ChatGPT to produce better answers
Outline

Introduction

Context

Contribution overview

Application to System Engineering

Evaluation

Demo

Conclusion
Contribution: Overview

Knowledge on system engineering

System specification

Diagram (SysML v2) Code

Question type: (Requirement identification, etc.)

AI Engine

Question

Answer

Automated analysis

Validated response

User interaction

TTool-AI
An Open-Source Toolkit

- Formal system modeling in SysML
  - Views: functional, architecture, deployment, system, etc.
  - Diagrams: requirement, use case, block, state machines, …
- Support for all system development cycle (from requirements to code)
- Safety / security / performance evaluation at the push of a button
- Code generation
- Industrial and academic contribution and use
- ttool.telecom-paris.fr
Contribution: Interacting with ChatGPT

Knowledge on system engineering

System specification

Diagram (SysML v2)

Code

Question type: (Requirement identification, etc.)

AI Engine

query

response

Question

Answer

Automated analysis

Validated response

User interaction

TTool-AI

apply
ChatGPT API: the Basic Way

```
$ curl https://api.openai.com/v1/chat/completions
-H "Content-Type: application/json"
-H "Authorization: Bearer OPENAI_API_KEY"
-d '{ "model": "gpt-3.5-turbo",
    "messages": [{"role": "user", "content": "Hello!"}]
}

{
    "id": "chatcmpl-7YzJ6Gmh0VjbWl3plUxqwzDrtDnnk",
    "object": "chat.completion",
    "created": 1688573172,
    "model": "gpt-3.5-turbo-16k-0613",
    "choices": [
        {"index": 0,
         "message": {"role": "assistant",
                      "content": "Hello! How can I assist you today?"},
         "finish_reason": "stop" }
    ],
    "usage": {"prompt_tokens": 8, "completion_tokens": 9,
              "total_tokens": 17 }
}
```

You first need to buy tokens to get a valid OPENAI_API_KEY
ChatGPT API: Still quite Basic

```
$ curl https://api.openai.com/v1/chat/completions
   -H "Content-Type: application/json"
   -H "Authorization: Bearer OPENAI_API_KEY"
   -d '{ "model": "gpt-3.5-turbo",
       "messages": [{"role": "user", "content": "Hello!"}]
   }' | jq '.choices[]'.message.content

Hello! How can I assist you today?

$ chatgpt "Is Supaero a good engineering school?"

"Supaero, also known as the Institut Superieur de l’Aeronautique et de l’Espace, is a renowned engineering school specializing in aerospace engineering in France. It consistently ranks among the top engineering schools globally and ..."
## ChatGPT API: Pricing

**Token-based**

- Input / output tokens
- Limits on the number of queries per unit of time

### ChatGPT 3.5

<table>
<thead>
<tr>
<th>Model</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>4K context</td>
<td>$0.0015 / 1K tokens</td>
<td>$0.002 / 1K tokens</td>
</tr>
<tr>
<td>16K context</td>
<td>$0.003 / 1K tokens</td>
<td>$0.004 / 1K tokens</td>
</tr>
</tbody>
</table>

### ChatGPT 4

<table>
<thead>
<tr>
<th>Model</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>8K context</td>
<td>$0.03 / 1K tokens</td>
<td>$0.06 / 1K tokens</td>
</tr>
<tr>
<td>32K context</td>
<td>$0.06 / 1K tokens</td>
<td>$0.12 / 1K tokens</td>
</tr>
</tbody>
</table>
Contribution: Custom Knowledge (or Context)

- Knowledge on system engineering
- System specification
- Diagram (SysML v2) Code
- Question type: (Requirement identification, etc.)

TTool-AI

- Automated analysis
- Query
- Response
- Question
- Answer
- Validated response
- User interaction
- Apply
ChatGPT API: Custom Knowledge (or Context) (Cont.)

$ curl https://api.openai.com/v1/chat/completions
...
"messages": [{"role": "user", "content": "Is Sophia-Antipolis sunny today?"}]
...
"I'm sorry, as an AI language model, I don't have access to current weather conditions. However, Sophia-Antipolis is located in the French Riviera, which has a generally sunny and Mediterranean climate..."
ChatGPT API: Custom Knowledge (or Context) (Cont.)

$ curl https://api.openai.com/v1/chat/completions \
... 
{"messages":[{"role":"user","content":"Today, in Sophia-Antipolis, the sun shines with no clouds."},{"role":"assistant","content":"ok"},
{"role":"user","content":"Is Sophia-Antipolis sunny today?"}]}
...

"Yes, you mentioned earlier that the sun is shining with no clouds in Sophia-Antipolis. So, it is indeed sunny today in Sophia-Antipolis. Enjoy the beautiful weather!"
Contribution: Automated Answer Analysis

Knowledge on system engineering

System specification

Diagram (SysML v2) Code

Question type: (Requirement identification, etc.)

AI Engine

query

response

Question

Answer

Automated analysis

Validated response

User interaction

TTool-AI

18/44
Automated Answer Analysis: Objectives

Invalid answer

Answer that does not respect the expected output:

- Invalid format (JSON, XML, SysML v2, ...)
- Constraints not respected
  - "List two blocks in json format", but the answer gives only one block

Detection and iteration

1. Detect format or constraint errors in answer
2. If at least one error is detected and \(cpt < maxIteration\) not reached
   2.1 The previous message and its answer are added to the context
   2.2 A description \(d_i\) is generated for each error \(e_i\)
   2.3 A new message is formed based on the list of \(d_i\)
   2.4 The message is fed into ChatGPT
   2.5 \(cpt++\); Goto 1
Automated Answer Analysis: Example

$ curl https://api.openai.com/v1/chat/completions

... 
"messages":[
{"role":"system",
 "content":"You are a helpful assistant for system engineering."},

{"role":"user",
 "content":"Give me the example of a SysML block, in json format, with attributes. No attribute of type String. The block should have one more attribute than ports, and one less port than operations. The number of ports must be equal to 1."}]

...
Automated Answer Analysis: Example (Cont.)

```json
{
  "name": "Block1", "attributes": [
    {"name": "attribute1", "type": "Integer"},
    {"name": "attribute2", "type": "Boolean"},
    {"name": "attribute3", "type": "Float"},
    {"name": "attribute4", "type": "Double"}
  ],
  "ports": [
    {"name": "port1", "type": "Port1"}
  ],
  "operations": [
    {"name": "operation1", "parameters": [], "returnType": "ReturnType1"},
    {"name": "operation2", "parameters": [], "returnType": "ReturnType2"}
  ]
}
```
Automated Answer Analysis: Example (Cont.)

We compute an updated question and we send it:

```bash
$ curl https://api.openai.com/v1/chat/completions \
... 
"messages": [
{"role":"system",
"content":"You are a helpful assistant for system engineering."},

{"role":"user",
"content":"Give me the example of a SysML block, in json format, with attributes. No attribute of type String.... The number of ports must be equal to 1."},

{"role":"assistant",
"content": { "name": "Block1", "attributes": [...]},

{"role":"user",
"content":"Your answer does not respect the following constraint: The block should have one more attribute than ports. Update your answer accordingly."}]
```

...
Automated Answer Analysis: Example (Cont.)

```json
{
    "name": "Block1", "attributes": [
        {"name": "attribute1", "type": "Integer"},
        {"name": "attribute2", "type": "Boolean"},
    ],
    "ports": [
        {"name": "port1", "type": "Port1"}
    ],
    "operations": [
        {"name": "operation1", "parameters": [], "returnType": "ReturnType1"},
        {"name": "operation2", "parameters": [], "returnType": "ReturnType2"}
    ]
}
```
Contribution: User Interaction and Result Displaying

Knowledge on system engineering
System specification
Diagram (SysML v2) Code
Question
Answer

AI Engine

TTool-AI

Automated analysis
User interaction

Validated response

Question type:
(Requirement identification, etc.)

System specification

Validated response

User interaction

Knowledge on system engineering

Diagram (SysML v2) Code

Question

Answer

TTool-AI

Automated analysis

Validated response

User interaction
Outline

Introduction

Context

Contribution overview

Application to System Engineering

Evaluation

Demo

Conclusion
General Contribution Applied to System Engineering

Objective: efficiently use the presented framework for automated system engineering

- Define a context (knowledge in system engineering, output format, expected output (a diagram, a list of elements, etc.))
- Define a list of questions linked with System Engineering
  - Requirement identification, requirement classification, ...
- Mandatory / optional inputs for questions:
  - System specification (text) - user provided
  - Diagram (in SysML V2 format)
- Automated analysis of answers, and iterations
- Diagram drawing from output
Focus on **SysML Design**

**SysML design**

= a set of interconnected blocks (structure) and one state machine for each block (behavior)

![SysML Diagram](image-url)
Focus on SysML Design: Knowledge Splitting

Context restriction

- Context = (Knowledge + question + specification) must fit in x tokens
- Providing knowledge on blocks, attributes, signals, connections, etc. take many tokens
  - ⇒ No space left for system specification and question
- Usually, large contexts also lead to worse results
- Price
Focus on SysML Design: Blocks

Context for the identification of blocks and attributes

When you are asked to identify SysML blocks, return them as a JSON specification formatted as follows:

```
{blocks: [{ "name": "Name of block", "attributes": [{"name": "name of attribute", "type": "int or bool" ...} ...]}
```

#Respect: each attribute must be of type "int" or "bool" only
#Respect: Any identifier (block, attribute, etc.) must no contain any space. Use "_" instead.
...
Focus on SysML Design: Blocks (Cont.)

Question for the identification of blocks and attributes

From the following system specification, using the specified JSON format, identify the typical system blocks and their attributes. Do respect the JSON format, and provide only JSON (no explanation before or after).

• Question is concatenated with the user-provided specification and user-provided (extra) constraints
• Automated verification: JSON format, constraints

Similar approach for:
• Identification of connections between blocks
• Identification of state machines
Focus on SysML Design: Blocks (Cont.)

Context for connecting blocks

...  
#Respect 2 signals with the same name are assumed to be connected: this is the only way to connect signals.  
#Respect: Two connected signals must have the same list of attributes, even if they are defined in two different blocks. One of them must be output, the other one must be input.  
#Respect: all input signals must have exactly one corresponding output signal, i.e., an output signal with the same name  
...

Question for connecting blocks

From the previous JSON and system specification, update this JSON with the signals you have to identify. If necessary, you can add new blocks and new attributes.
## Outline

<table>
<thead>
<tr>
<th>Introduction</th>
<th>Context</th>
<th>Contribution overview</th>
<th>Application to System Engineering</th>
<th>Evaluation</th>
<th>Demo</th>
<th>Conclusion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Introduction
- Context
- Contribution overview
- Application to System Engineering
- Evaluation
- Demo
- Conclusion
TTool-AI vs Master-Level Students

Test setup

- Same system specification is given to TTool-AI and to 15 master-level students after 21h of training
- TTool-AI’s models and Student’s models are evaluated with the same grading policies
- Comparison on average and standard deviation for grades, and for the time taken to produce models
**TTool-AI vs Master-Level Students (Cont.)**

<table>
<thead>
<tr>
<th></th>
<th>Time BD (s)</th>
<th>Grade BD (/100)</th>
<th>Time SMD (s)</th>
<th>Grade SMD (/100)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average</strong></td>
<td>40</td>
<td>81</td>
<td>178</td>
<td>63</td>
</tr>
<tr>
<td><strong>Std dev.</strong></td>
<td>10</td>
<td>16</td>
<td>97</td>
<td>15</td>
</tr>
<tr>
<td>(a) <strong>TTool-AI</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Time BD (s)</th>
<th>Grade BD (/100)</th>
<th>Time SMD (s)</th>
<th>Grade SMD (/100)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average</strong></td>
<td>2700</td>
<td>70</td>
<td>2700</td>
<td>58</td>
</tr>
<tr>
<td><strong>Std dev.</strong></td>
<td>—</td>
<td>26</td>
<td>—</td>
<td>32</td>
</tr>
<tr>
<td>(b) <strong>Students</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Better average for TTool-AI on grades, and it is much faster
- Best students always do better than TTool-AI, but with a much higher time
  - And it is hard to predict which students will be the best for a given specification
Demo: a Coffee Machine

Objective: making the block and state machine diagrams of the following system:

This coffee machine delivers a beverage once two coins have been inserted. If the delay between the first coin and the second coin is too long, the first coin is ejected. Similarly, if the delay to select a beverage is too long once the two coins have been inserted, the two coins are ejected. If one of the button (tea, coffee) is pressed before coins are ejected, the beverage is prepared. Once the beverage has been retrieved, the machine can accept new coins.

TTool is used for query/answer to/from ChatGPT and diagram drawing ttool.telecom-paris.fr
Demo: a Coffee Machine (Cont.)

This coffee machine delivers a beverage once two coins have been inserted. If the delay between the first coin and the second coin is too long, the first coin is ejected. Similarly, if the delay to select a beverage is too long once the two coins have been inserted, the two coins are ejected. If one of the button (tea, coffee) is pressed before coins are ejected, the beverage is prepared. Once the beverage has been retrieved, the machine can accept new coins.
Demo: a Coffee Machine (Cont.)
Demo: a Coffee Machine (Cont.)

block CoffeeMachine
- coinInserted = false : bool;
- buttonPressed = false : bool;
- delayBetweenCoins = 0 : int;
- delayToSelectBeverage = 0 : int;
- beveragePrepared = false : bool;
- in insertCoin(int coinValue)
- in pressButton(int buttonType)
- out beverageReady0
- out ejectCoins()

block Coin
- value = 0 : int;
- out insertCoin(int coinValue)

block Beverage
- in beverageReady0

block Button
- type = 0 : int;
- out pressButton(int buttonType)
Demo: a Coffee Machine (Cont.)

This coffee machine delivers a beverage once two coins have been inserted. If the delay between the first coin and the second coin is too long, the first coin is ejected. Similarly, if the delay to select a beverage is too long once the two coins have been inserted, the two coins are ejected. If one of the buttons (tea, coffee) is pressed before coins are ejected, the beverage is prepared. Once the beverage has been retrieved, the machine can accept new coins.
Demo: a Coffee Machine (Cont.)
Outline

Introduction

Context

Contribution overview

Application to System Engineering

Evaluation

Demo

Conclusion
To Go Further with TTool-AI... 

ChatGPT is a chatbot: it tells stories related to your questions, elaborating a probable answer, with no guarantees that the answer is correct.

1. **Analysis of system specification is rather impressive**: extracting requirements, use cases, design (structure, behavior) and properties identification

2. **Need for automated and manual supervision**
   - Trade-off between quality, price and time

3. **Best students can do better but with a much higher time**
   - Trade-off between quality and time, and how to pre-identify best students?

4. **Verifying that a property is true for a given system usually fails**
   (ChatGPT can be wrong at solving trivial mathematical exercises...)
   - ChatGPT refuses to prove properties for complex diagram

Version 4 of ChatGPT is even better, not covered by this presentation
Questions?

Latest version of TTool includes what has been demonstrated... and much more!
ttool.telecom-paris.fr