LLM Based SysML Virtual Assistant

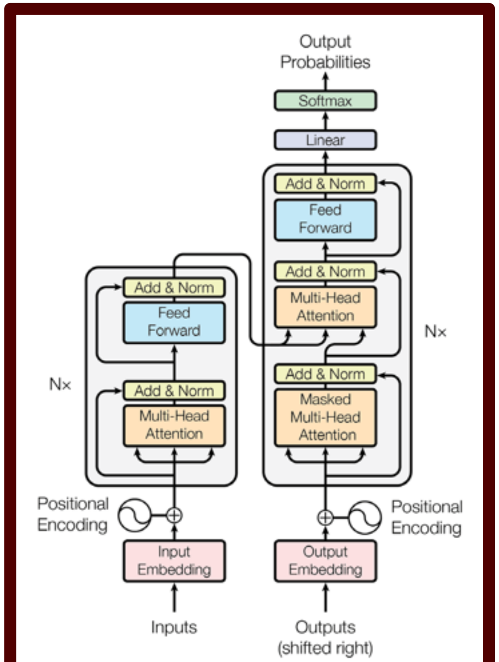
Alex Demagall, Gabe Apaza, Daniel Selva
SEAK Lab

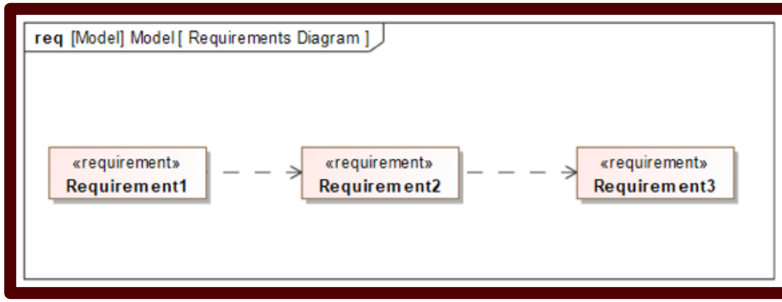
9/27/2023
AI4SE & SE4AI Workshop

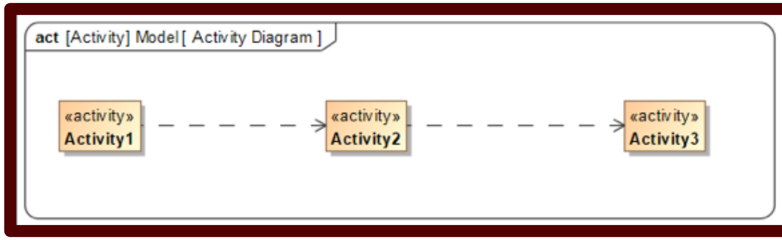
Background

- Large Language Models (LLMs) have been shown to have a wide range of applications outside of text generation.

- Systems Modeling Language (SysML) is a modeling language for systems engineering applications.



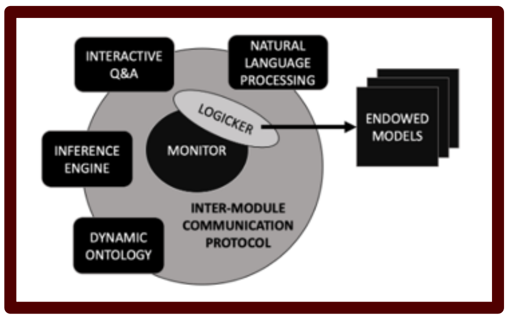




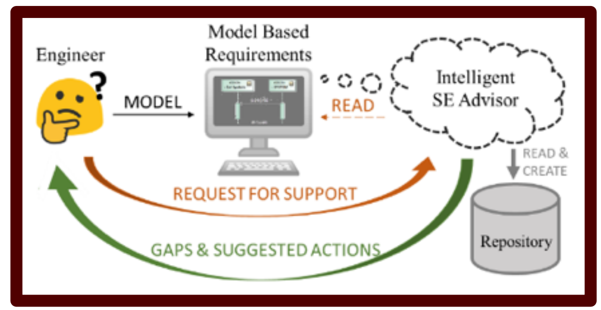
Lit Review and Research Goal

- A virtual assistant to perform simple tasks and shorten the learning curve for SysML software is desired.

- Similar past work has focused on information management (SEVA) [Kri. 2019] or requirements completion [Sal. 2020]. This work focuses on model creation and analysis.



SEVA System Concept



Requirements Completion Concept

- Research Question: How can we use LLMs to help users create better SysML models faster?

Specific Objectives

- Create a virtual assistant for SysML software with three functionalities.

Read a SysML Project

Edit a SysML Project

Answer General Questions

- Integrate the virtual assistant into a commonly used systems modelling software.

- Employ these functionalities using natural language input from the user.

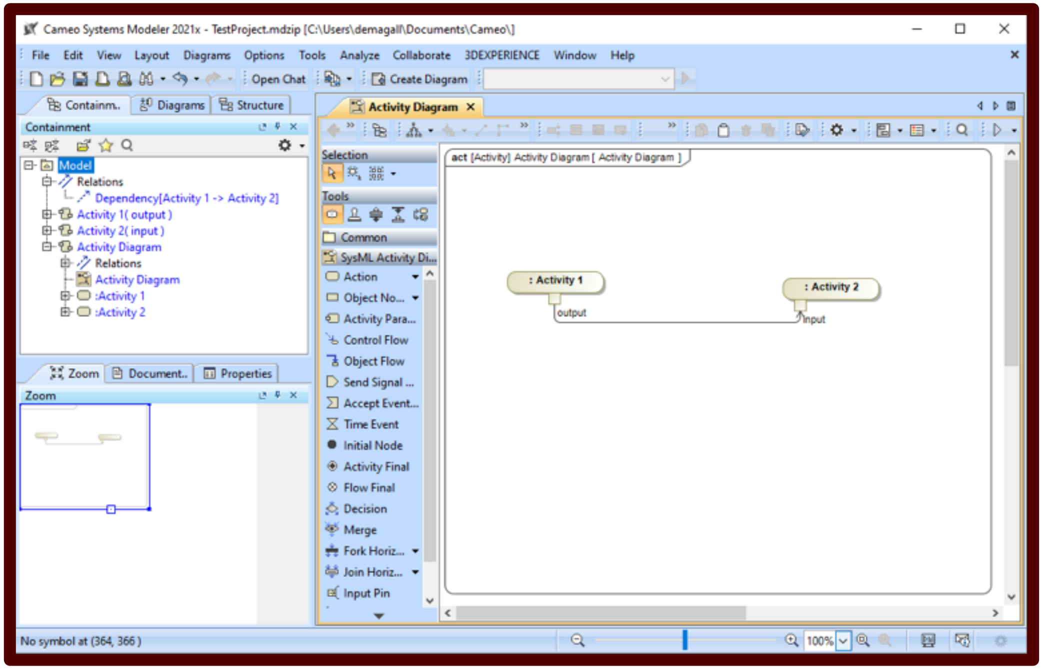
Approach

- The virtual assistant was created as a plugin for Cameo Systems Modeler.

- The plugin integrated OpenAI's GPT-4 as the LLM base for the assistant.

Cameo

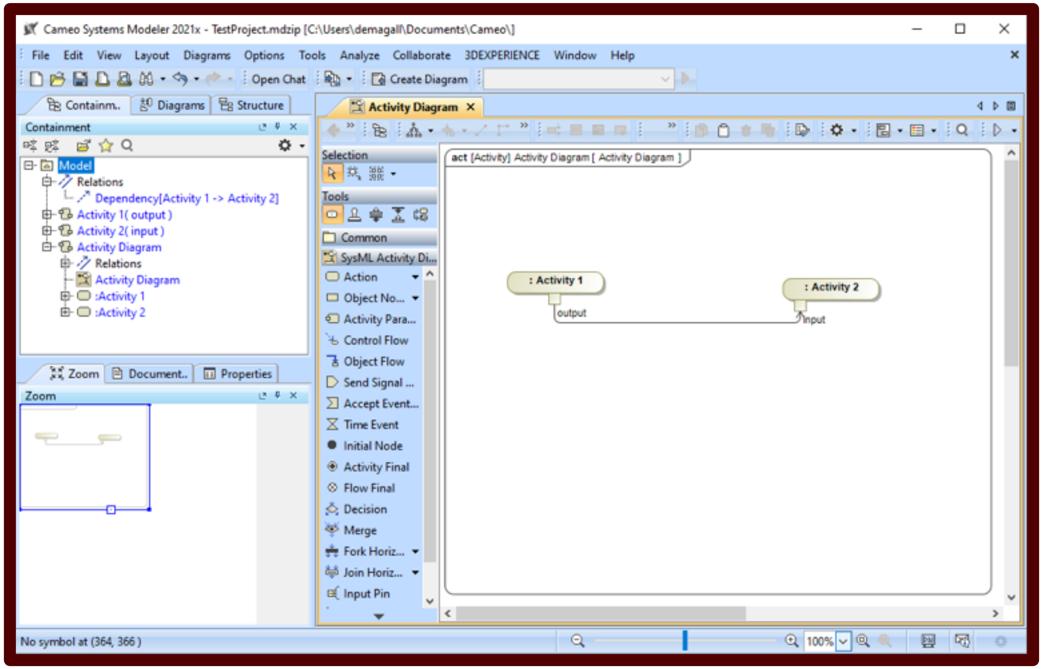
- Cameo Systems Modeler is a commonly used tool for systems engineers to perform model based systems engineering, especially with the SysML modeling language.
- Was chosen for its widespread use in the industry.



Cameo Terms

- **Element**: Any ‘thing’ in the model. Models, classes, diagrams, and relationships are all types of elements.

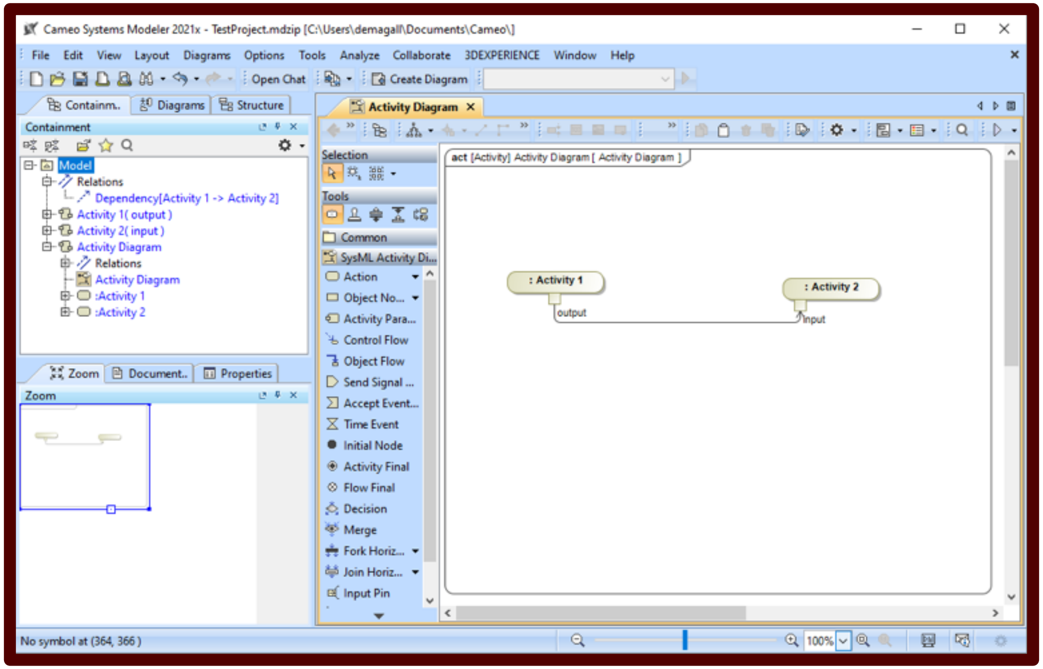
- **Model**: Can be the parent model or a sub model. Generally used to contain a collection of diagrams, relationships, and other elements.



Cameo Terms

- **Diagram**: Can display elements and the relationships between them.

- **Relationship**: Establishes a connection between any two elements.



GPT-4 Integration

- GPT-4 is integrated into the plugin using a chat window.
- LLMs have been shown to learn with few examples. [Bro. 2020]
  - This was done through in-context learning
  - Only 8 examples were given to GPT-4 for this project.
- This was enough for the tool to accurately write in the format needed by the parser.
- GPT-4's base knowledge was used, no extra knowledge was added.

Parser

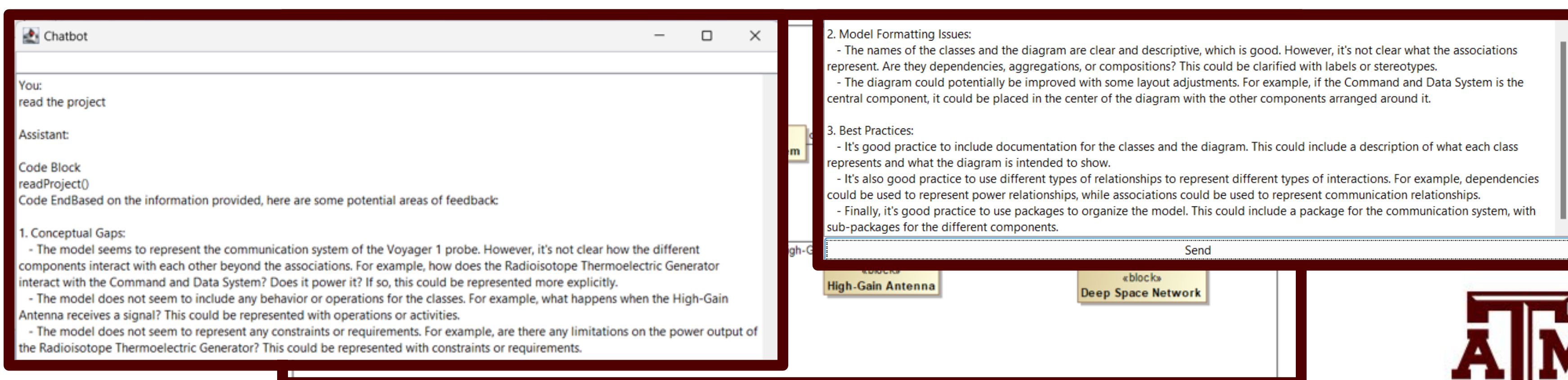
- The parser is activated by the phrase "Code Block" and deactivated by the phrase "Code End".

- In-between those phrases, the parser looks for five possible functions:
  - createDiagram()
  - addElement()
  - addRelationship()
  - removeElement()
  - readProject()

- Any needed arguments were formatted in the following format:
  - Name 'nameofelement'
  - Type 'typeofelement'
  - Owner 'nameofowner'

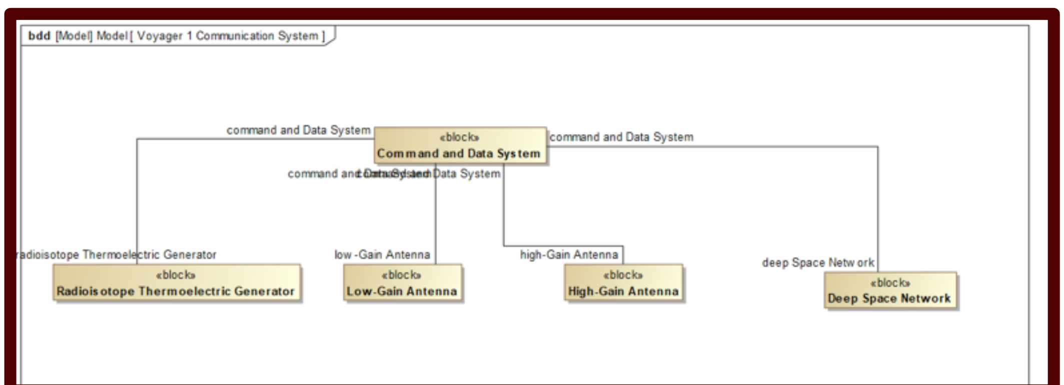
Read SysML Project

- The plugin iteratively reads through the project and collects information about each element (such as name, type, owner).
- This information is transformed into natural language and passed to the LLM.
- GPT-4, using prior knowledge about SysML gives feedback about conceptual gaps, model formatting issues, and best practices.



Read Project Examples

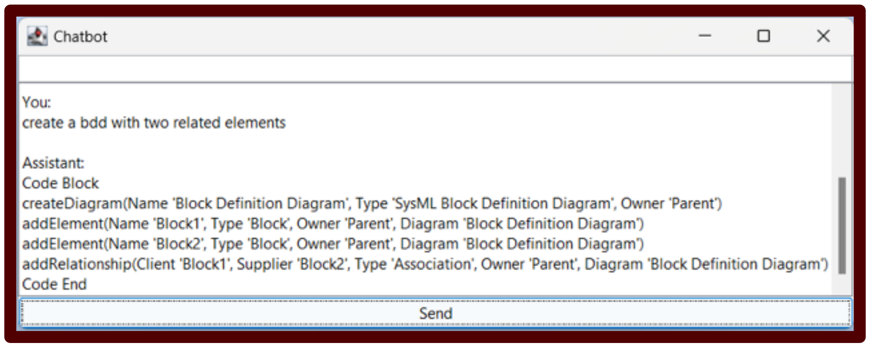
- Conceptual Gaps
  - "It's not clear how the different components interact with each other beyond the associations. For example, how does the Radioisotope Thermoelectric Generator Interact with the Command and Data System? Does it power it? If so, this could be represented more explicitly"
  - "The model does not seem to include any behavior or operations for the classes. For example, what happens when the High-Gain Antenna receives a signal? This could be represented with operations or activities."
- Model Formatting Issues
  - "It's not clear what the associations represent. Are they dependencies, aggregations, or compositions? This could be clarified with labels or stereotypes."
- Best Practices
  - "It's good practice to include documentation for the classes and the diagram. This could include a description of what each class represents and what the diagram is intended to show."



Edit SysML Project

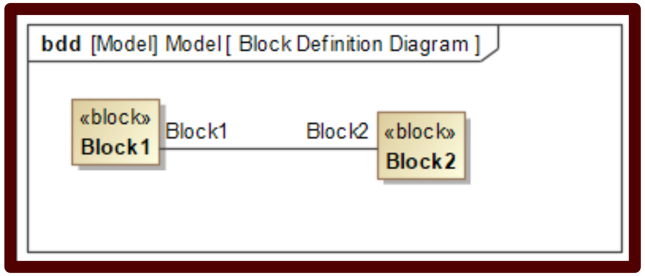
- The user gives the LLM a command in natural language. The LLM classifies the command into several distinct tasks, each of which can be performed using a function.

- The LLM responds with a carefully formatted message which calls a parser, which runs the desired function with the desired arguments.



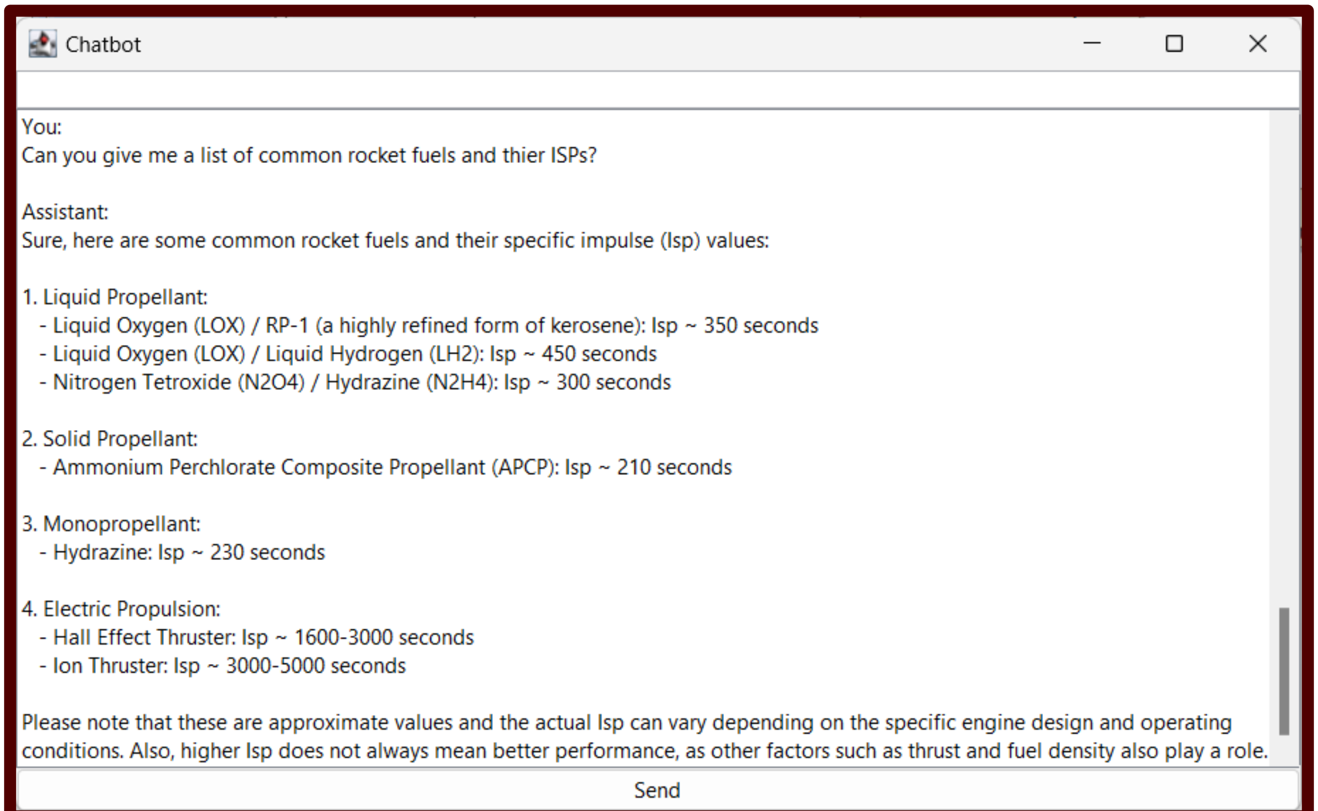
You:
Create a bdd with two related elements

Assistant:
createDiagram(Name ‘Block Definition Diagram’,
Type ‘SysML Block Definition Diagram’, Owner ‘Parent’)



Answer Questions

- If no specific task is recognized, the LLM will revert to GPT-4's standard behavior.



Limitations

- Hallucinations
  - The program will often “invent” new functions or element types which are not in the examples provided to it.
  - This means that either the parser can’t recognize it or cameo can’t recognize it.
- Functions in parser
  - Not everything that can be done in Cameo is in the parser (comments, labels, attributes, etc.).
- Feedback is ok, but repetitive
  - It always follows the same format.
  - It tends to catch the same kinds of errors

Future Work

- Expand the functions the parser can recognize.
- Improve feedback
  - Better examples
  - Add domain specific knowledge
- More examples may reduce hallucinations, however this is a common problem in LLMs with lots of active research.
- User validation is needed.

References

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). *Language Models are Few-Shot Learners* (arXiv:2005.14165). arXiv. https://doi.org/10.48550/arXiv.2005.14165

Krishnan, J., Coronado, P., & Reed, T. (2019). *SEVA: A Systems Engineer's Virtual Assistant*.

Salado, A. (2020). From Model-Based Requirements to a Virtual Systems Engineering Advisor that Identifies Gaps in Requirements: An Application to Space Systems. In *ASCEND 2020*. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2020-4092

Tarassow, A. (2023). *The potential of LLMs for coding with low-resource and domain-specific programming languages* (arXiv:2307.13018). arXiv. http://arxiv.org/abs/2307.13018

Wang, B., Wang, Z., Wang, X., Cao, Y., Saurous, R. A., & Kim, Y. (2023). *Grammar Prompting for Domain-Specific Language Generation with Large Language Models* (arXiv:2305.19234). arXiv. http://arxiv.org/abs/2305.19234