NAVAL POSTGRADUATE SCHOOL

Leveraging Generative AI to Build, Modify, and Query MBSE Models

Acquisition Research Symposium

8 May 2024

Ryan Longshore
Ryan Bell
Raymond Madachy, PhD

Agenda

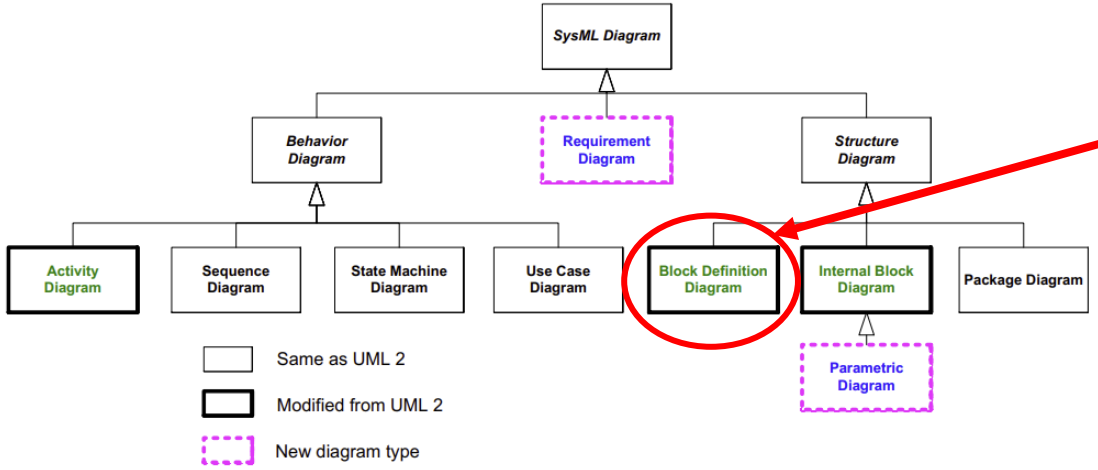
- Systems Modeling
- Large Language Models (LLM)
- Improving LLM Responses
- Custom GPTs
- Bicycle Case Study
- Future Research

Research Question

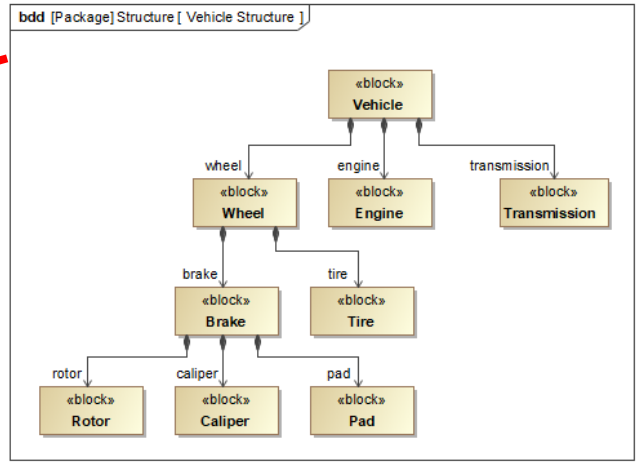
Can Large Language Models (LLMs) be used to generate, modify, and query systems models?

Systems Modeling

- Model Based Systems Engineering (MBSE) – "A Formalized application of modeling to support system requirements design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later lifecycle phases" *(INCOSE, 2007)*
- Systems Modeling Language (SysML) developed to implement a MBSE methodology.
  - SysML v1.x is a graphical programming language that is an extension of the Unified Modeling Language (UML) version 2 Standard with two new diagram types



SysML v1 Diagram Taxonomy *(OMG, 2017)*



Vehicle BDD *(NoMagic)*

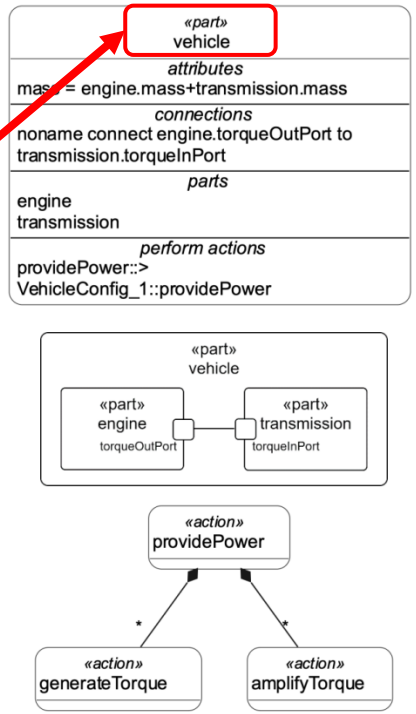
Systems Modeling

- SysML v2 – June 2023 Beta Release. Key Differences *(Friedenthal, 2023)*:
  - New metamodel: Extends the Kernel Modeling Language (KerML), not UML
  - Standard API
  - **Textual Notation** in addition to Graphical Notation

```
part vehicle{
    attribute mass = engine.mass+transmission.mass;
    perform providePower;
    part engine{
        attribute mass;
        port torqueOutPort;
        perform providePower.generateTorque;
    }
    part transmission{
        attribute mass;
        port torqueInPort;
        perform providePower.amplifyTorque;
    }
    connect engine.torqueOutPort to transmission.torqueInPort;
}
action providePower{
    action generateTorque;
    action amplifyTorque;
}
```

Parts Introduced in SysML 2

No Blocks in SysML 2

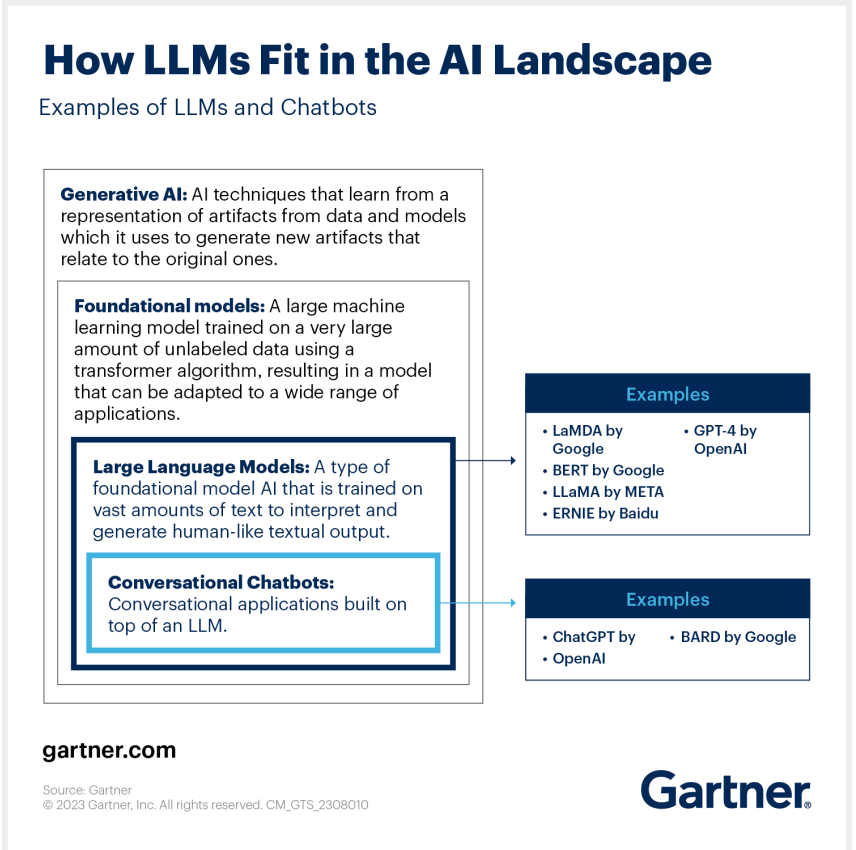


Vehicle Textual Notation Model
(Friedenthal, 2023)

Vehicle Graphical Notation Model
(Friedenthal, 2023)

Large Language Models (LLMs)

- Deep learning natural language processing (NLP) algorithms trained on large datasets
- Generate human-like text based on natural language prompts
- Chatbot interface
- Some are multi-modal (input/output images, sound, or sound in addition to text)
- Use cases include Code Generation and Modification



Improving LLM Responses

- **Re-Training:** Re-train base model with new dataset
- **Finetuning:** Teach the base model specialized knowledge by adjusting a small number of internal parameters. Significant time savings compared to re-training entire model (*Nucci, 2024*)
- **Retrieval Augmented Generation (RAG):** Utilize external data to augment the LLM's knowledge without changing model weights (*Nucci, 2024*)
- **Prompt Engineering:** Optimize prompts for optimal outputs
  - **Few Shot** prompts teach LLMs to perform a task through examples (*Oleszak, 2024*)
  - **Zero Shot** prompts challenge the LLM to perform a task correctly when it has not been correctly trained for that task (*Oleszak, 2024*)

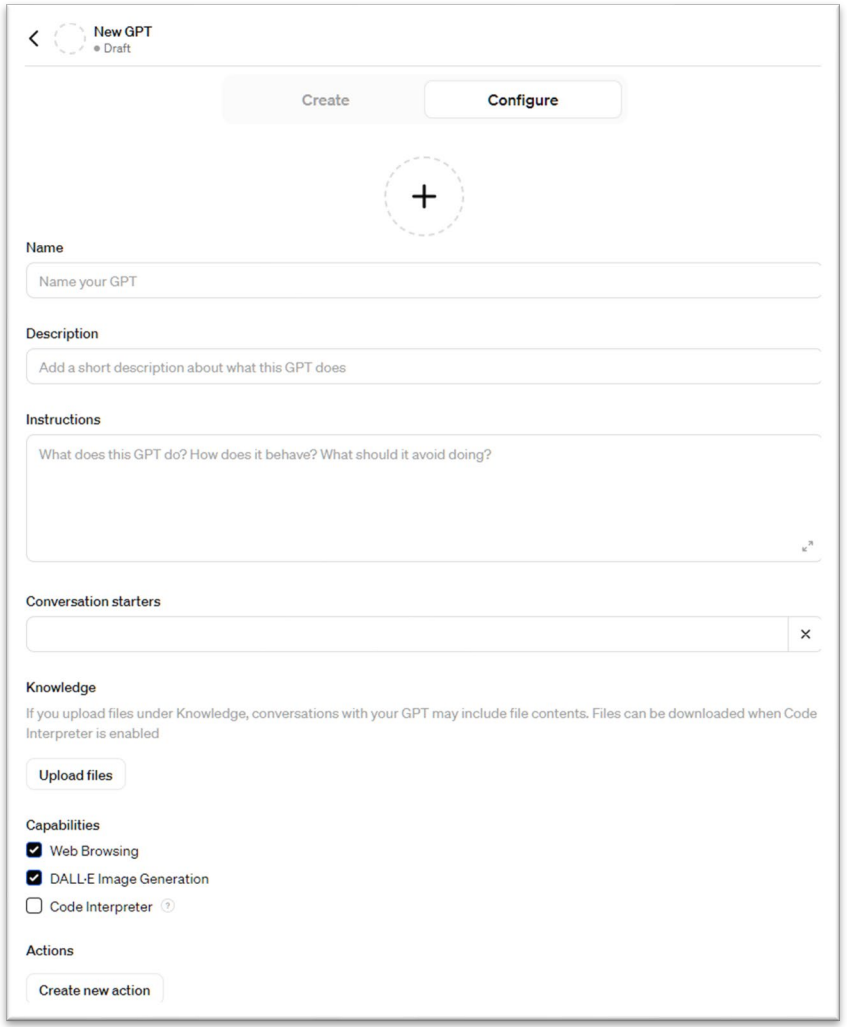
These methods can also be combined.

Custom GPT: Senior Systems Engineer – Systems Modeler (SSE-SM)

OpenAI's custom GPT feature enables users to build custom GPTs by giving GPT-4 additional knowledge in the form of custom instructions, recommended prompts (conversation starters), and knowledge (RAG).

SSE-SM is a custom GPT focused on SysML v2 modeling. It contains:

- Custom instruction set related to modeling
- Domain Specific Knowledge
  - SysML v2 specification and notation documents from SysML v2 GitHub
  - SysML v2 template
  - Se-lib.py Python Library



Bicycle Case Study Overview

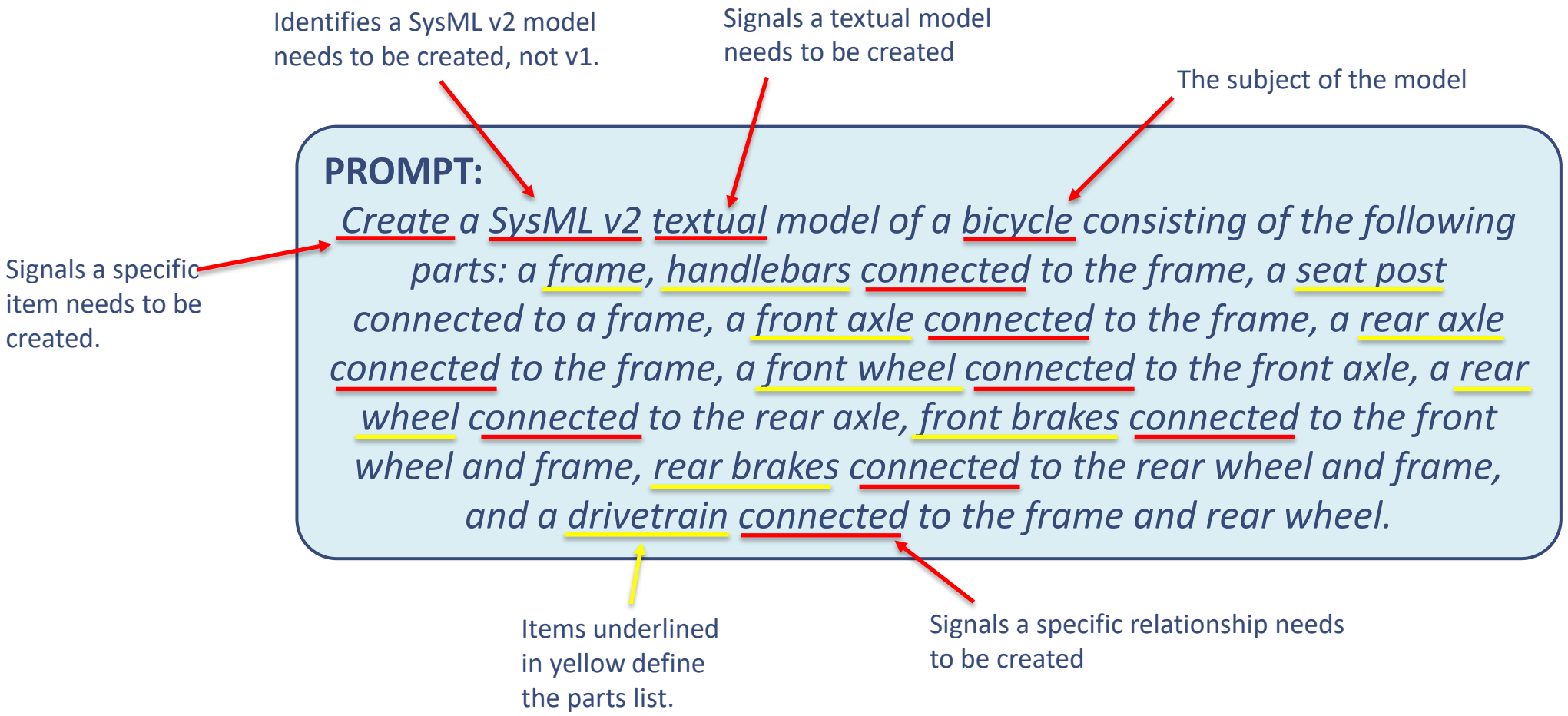
Question 1: Can we utilize a LLM to build a SysML v2 textual notation model of a bicycle using natural language prompts?

Question 2: Can we utilize a LLM to modify a SysML v2 textual notation model of a bicycle using natural language prompts?

LLMs Utilized:

- ChatGPT-3.5
- ChatGPT-4
- Custom GPT: Senior Systems Engineer – Systems Modeler (SSE-SM)

Bicycle Case Study: Create Prompt



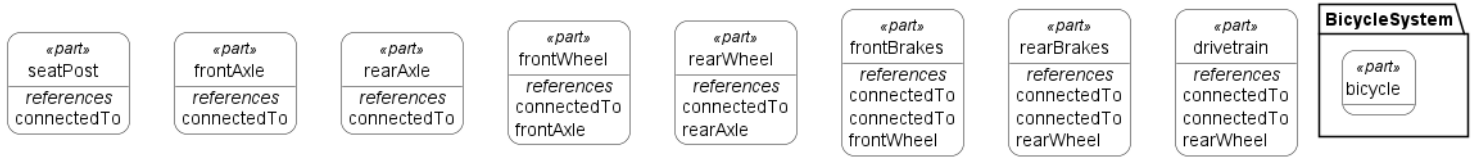
Bicycle Case Study: Create with ChatGPT-3.5

Zero Shot Prompt: Not Successful

Few Show Prompt: Able to correct some of the syntax, but valid model was not successful

```
package BicycleSystem {

    part bicycle: Bicycle {
        frame,
        handlebars,
        seatPost,
        frontAxle,
        rearAxle,
        frontWheel,
        rearWheel,
        frontBrakes,
        rearBrakes,
        drivetrain
    }

    part frame: Frame

    part handlebars: Handlebars {
        connectedTo(frame)
    }

    part seatPost: SeatPost {
        connectedTo(frame)
    }
// END OF SAMPLE CODE
}
```

Zero Shot Model



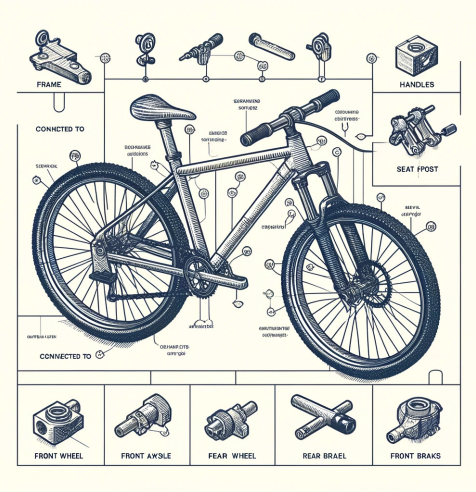
Few Shot Model

Bicycle Case Study: Create with ChatGPT-4

Zero Shot Prompt: Limited Success

Few Show Prompt: Successful

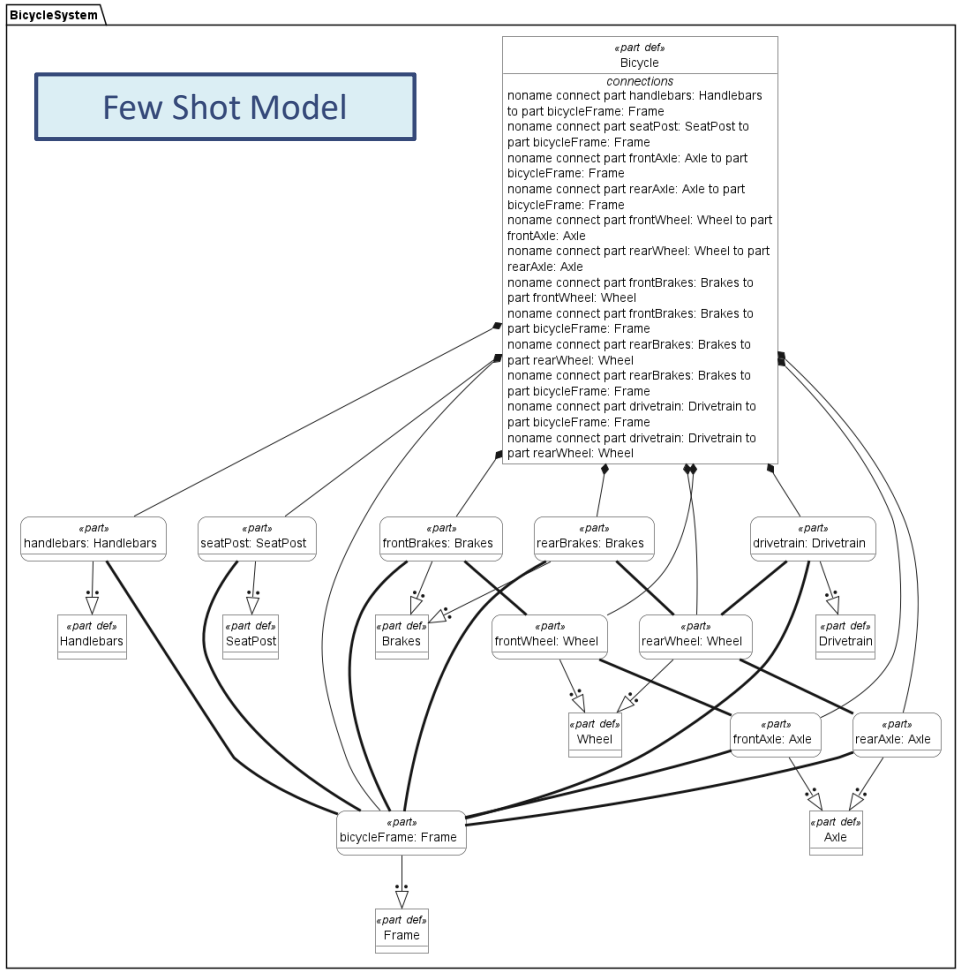
When "textual" was not originally included in the prompt, ChatGPT-4 called Dall-E to create a graphical model. This was corrected in the latest release of ChatGPT-4.




```
package BicycleSystem {
// SAMPLE CODE BEGINS
    part def Brakes {
    }

    part def Drivetrain {
    }

    part def Bicycle {
        part frame: Frame;
        part handlebars: Handlebars;
        part seatPost: SeatPost;
        part frontAxle: FrontAxle;
        part rearAxle: RearAxle;
        part frontWheel: Wheel;
        part rearWheel: Wheel;
        part frontBrakes: Brakes;
        part rearBrakes: Brakes;
        part drivetrain: Drivetrain;

        // Connections
        connect frame -> handlebars;
        connect frame -> seatPost;
        connect frame -> frontAxle;
        connect frame -> rearAxle;
        connect frontAxle -> frontWheel;
        connect rearAxle -> rearWheel;
        connect frame -> frontBrakes -> frontWheel;
        connect frame -> rearBrakes -> rearWheel;
        connect frame -> drivetrain -> rearWheel;
    }
}
```

Zero Shot Model

Few Shot Model

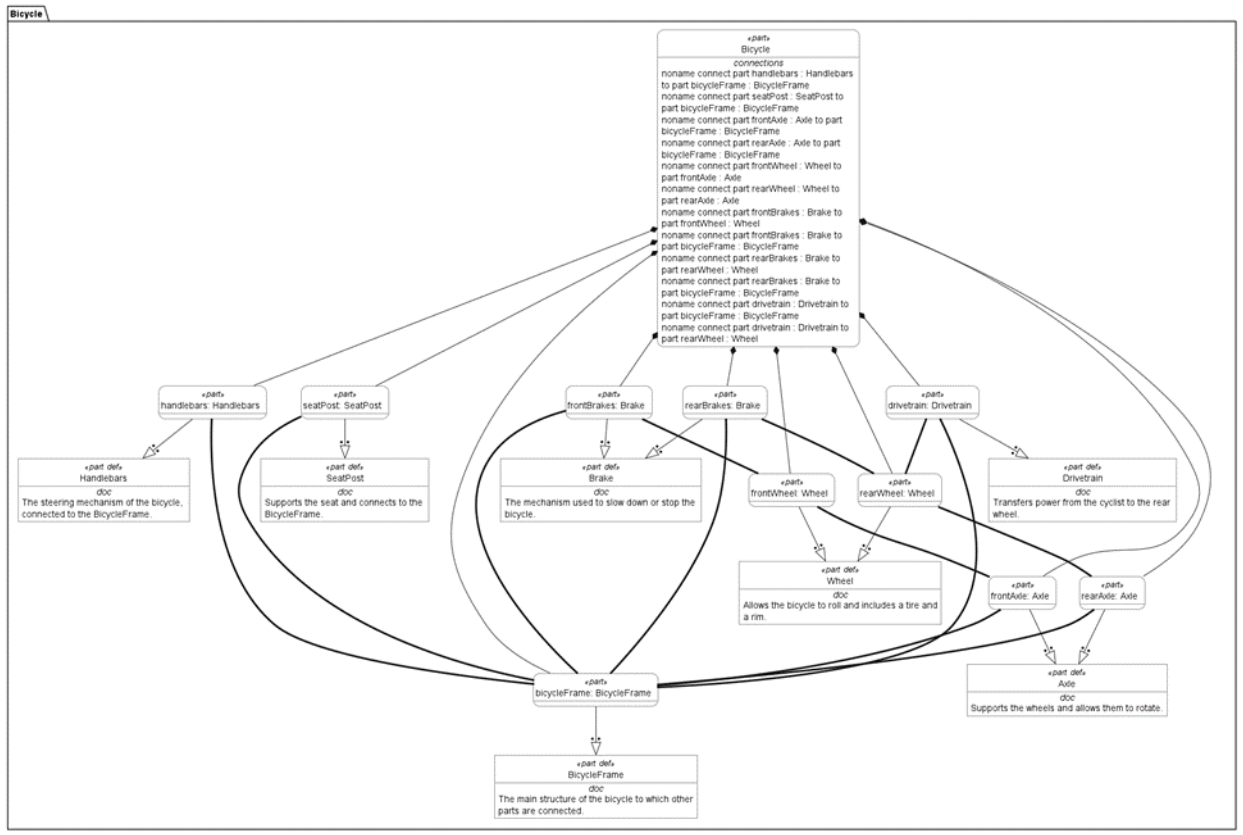


Bicycle Case Study: Create with SSE-SM

Case study adapted from previous research (Longshore, in press)

Initial Prompt was **successful**. Note that this is not a zero shot prompt as the LLM was given additional domain specific information.

```
package Bicycle {

    // Begin Sample Code

    part def Brake {
        doc /* The mechanism used to slow down or stop the bicycle. */
    }

    part def Drivetrain {
        doc /* Transfers power from the cyclist to the rear wheel. */
    }

    part Bicycle {
        part bicycleFrame : BicycleFrame;
        part handlebars : Handlebars;
        part seatPost : SeatPost;
        part frontAxle : Axle;
        part rearAxle : Axle;
        part frontWheel : Wheel;
        part rearWheel : Wheel;
        part frontBrakes : Brake;
        part rearBrakes : Brake;
        part drivetrain : Drivetrain;

        // Connections
        connect handlebars to bicycleFrame;
        connect seatPost to bicycleFrame;
        connect frontAxle to bicycleFrame;
        connect rearAxle to bicycleFrame;
        connect frontWheel to frontAxle;
        connect rearWheel to rearAxle;
        connect frontBrakes to frontWheel;
        connect frontBrakes to bicycleFrame;
        connect rearBrakes to rearWheel;
        connect rearBrakes to bicycleFrame;
        connect drivetrain to bicycleFrame;
        connect drivetrain to rearWheel;
    }
}
```



Bicycle Case Study: Modify Prompt

Model created by SSE-SM in the Build Phase effectively giving it an example to work from for adding these two parts.

PROMPT:
Modify the following SysML v2 model to add a left pedal to the drivetrain and a right pedal to the drivetrain:

[SysML v2 Model]

New Parts

PROMPT:
Modify the model to include the following requirement:
"The bike shall support a user weighing 300 pounds or less."

Bicycle Case Study: Modifying the Model

Requirements Modification

GPT-3.5: Modeled as a Constraint and not Requirement

```
part def BicycleFrame {
    doc /* The main structure of the bicycle to which other parts are connected. */
    constraint maxUserWeight: Real[1] { max <= 300 } // Constraint for maximum user weight
}
```

GPT-4: Requirement Modeled Correctly

```
requirement def WeightCapacity {
    doc /* The bike shall support a user weighing 300 pounds or less. */
}
```

SSE-SM: Requirement Modeled Correctly. Also added a requirement number

```
requirement def <'1'> UserWeightSupport {
    doc /* The bike shall support a user weighing 300 pounds or less. */
}
```

Pedal Modification: All LLMs implemented different, but valid, ways of making the pedal modification.

Bicycle Case Study: Querying the Model

- Simple Recall: *How many brakes the bicycle does the bicycle contain?*
  - All LLMs correctly identified two brakes, front and rear.
- *Will the bicycle be able to stop if the rear brakes fail?*
  - All LLMs correctly identified the bicycle would be able to stop using the front brakes.
- *Will the bicycle be able to stop if the front brakes fail?*
  - All LLMs correctly identified the bicycle would be able to stop using the rear brakes.
- *Will the bicycle be able to stop if both brakes fail?*
  - All LLMs identified this would be a failure of the braking system and gave alternate options for stopping the bicycle. All LLMs also identified the braking system was a critical system for the bike and recommended regular inspection and maintenance.

Future Research

- **AI Assistance Cost Factors:** Inform cost factors by capturing metrics related to using general and domain specific LLMs (Madachy, in press)
- **Systems Modeling Domain Specific LM:** Deep expertise/understanding of Systems Modeling. May also be smaller, requiring less memory and compute enabling LMs to be run locally.
- **Systems Modeling Benchmark:** Quantify LLMs capabilities to perform systems modeling functions. Build upon SysEngBench (Bell, in press).
- **SysML v2 Model Library:** Curated SysML v2 model dataset inclusive of all aspects of the language to be used training and finetuning LMs. (Supports previous two bullets)
- **Model Conversion:** Explore the ability of LLMs to convert SysML v1.x models to SysML v2.

Conclusion

As LLMs improve, they show an increasing ability to build and modify systems models. There is great potential to increase modeling efficiency and enhance the practice of systems engineering. Continued research is needed to optimize results and accurately measure performance.

References

- Bell, Ryan, Raymond Madachy, and Ryan Longshore. “Introducing SysEngBench: A Novel Benchmark for Assess Large Language Models in Systems Engineering.” *Proceedings of the 2024 Acquisition Research Symposium*. Monterey, CA, (in press).
- “Defining Blocks in Block Definition Diagram - SysML Plugin 19.0 LTR - No Magic Documentation.” Accessed April 12, 2024. https://docs.nomagic.com/display/SYSMLP190/Defining+Blocks+in+Block+Definition+Diagram.
- Friedenthal, Sanford, and Ed Seidewitz. “SysML v2: Highlighting the Differences with SysML V1.” *PPI SyEN*, no. 123 (April 2023): 35–55.
- “INCOSE Systems Engineering Vision 2020.” INCOSE, September 2007. https://sdincose.org/wp-content/uploads/2011/12/SEVision2020_20071003_v2_03.pdf.
- Longshore, Ryan, Raymond Madachy, and Ryan Bell. “Leveraging Generative AI to Create, Modify, and Query MBSE Models.” *Proceedings of the 2024 Acquisition Research Symposium*. Monterey, CA, (in press).
- Madachy, Raymond, Ryan Bell, and Ryan Longshore. “Systems Acquisition Cost Modeling Initiative for AI Assistance.” *Proceedings of the 2024 Acquisition Research Symposium*. Monterey, CA, (in press).
- Nucci, Antonio. “RAG vs Fine-Tuning LLM: Comparing the Gen AI Approaches.” *Aisera: Best Generative AI Platform For Enterprise* (blog), January 20, 2024. https://aisera.com/blog/llm-fine-tuning-vs-rag/.
- Oleszak, Michał. “Zero-Shot and Few-Shot Learning with LLMs.” neptune.ai, March 22, 2024. https://neptune.ai/blog/zero-shot-and-few-shot-learning-with-llms.
- “OMG Systems Modeling Language Version 1.5,” May 2017. http://www.omg.org/spec/SysML/1.5.