TANGRAMFLEX®

LLM Co-pilots for Domain Specific Modeling Languages

AI4SE & SE4AI Research and Application Workshop
September 18, 2024

Matt Naveau, Chief Technology Officer
matt.naveau@tangramflex.com

Some Baseline Terminology

- **Artificial Intelligence (AI)** - *computer systems performing tasks that normally require human intelligence*
- **Generative AI** - *AI capable of generating text, images, or other data using generative models, often in response to prompts*

- **Large Language Models (LLMs)** - *AI that is capable of general-purpose language generation*
- **GPT** - *A general purpose LLM created by OpenAI*
- **ChatGPT** - *A user-friendly chat prompt interface that uses GPT created by OpenAI*

System Models and Generative AI in DoD Context

- All of the services are pushing to design, develop, test, and deploy faster
- Generative AI, which burst into mainstream consciousness in late 2022, has the potential to make lots of things go faster
- All of the services are pushing for increased digital engineering, digital transformation, and government-owned architectures and interfaces

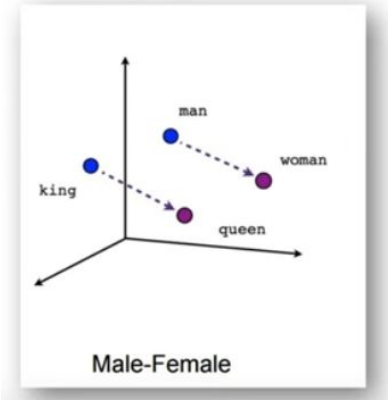
As a DoD industry, we need a way to "go faster" and lean into digital engineering, and generative AI is a key enabling technology to make this happen

Generative AI for Making System Models

- Currently available generative AI technology is great at producing text for paper writing, emails, proposal writing, etc.
- Generative AI also holds great potential for accelerating digital systems modeling
  - When used effectively, generative AI can significantly reduce the human labor required to create digital models
- To fully achieve the promise, generative AI must first overcome significant scaling challenges

Generative AI for System Models - Scaling Challenges

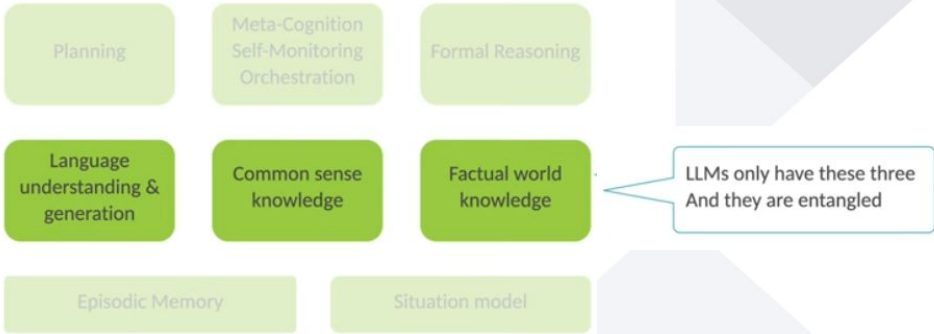
- Current generative AI technology suffers from a correctness/accuracy problem
  - LLMs can't tell if what they've produce is correct or not
  - Incorrect outputs are commonly called "Hallucinations"
  - This is due to the way LLMs work



The geometry of words

Dr. John Launchbury
Tech Talk - "The Trajectory of AI"

https://www.youtube.com/watch?v=OPFSXSZmeOQ



The elements of human intelligence

Generative AI for System Models - Scaling Challenges

- The hallucination problem is compounded when generating general purpose modeling specification
  - A wide-open specification gives the LLM many more options to generate results that may or may not be correct
  - Checking the results is usually a manual and tedious process
- Generative AI is most effective at producing text
  - Not all commonly used modeling tools adhere to a semantically-rigid textual standard, making it more difficult for the AI to produce accurate models

Domain Specific Modeling Languages (DSMLs) & Generative AI

What is a Domain Specific Modeling Language (DSML)?

- DSML is a language designed to model within a constrained domain
  - Intentionally is not general purpose
  - Can’t model everything
- DSMLs typically have tighter syntax than general purpose languages
  - Well suited for precise specification enabling mathematical analysis, automatic code generation, configuration of systems, etc.
- DSMLs are increasingly used in industry and the DoD
  - Examples: VHDL, Terraform, YAML for CI/CD pipelines, MATLAB, AADL, Cryptol for cryptography, Flex for integration modeling

DSMLs Help Address LLM Scaling Challenges

- DSMLs are text based, making them ideal targets for generative AI
- LLMs are straightforward to train to produce syntactically accurate DSML code
  - DSMLs require less simultaneous context for the LLM than a broader general purpose language
  - Retrieval Augmentation Generation (RAG) can effectively augment general purpose LLMs without special model training
- LLMs can be augmented with tools built to work on DSMLs
  - Example: LLM could directly interact with a syntax checker and iterate until it has produced fully valid code

Examples From Our Work

Some Examples From Our Work

- Using generative AI to produce Flex, an integration modeling language, to create digital representations of message standards for code generation
- Using generative AI to produce a Tangram Pro component model that was automatically converted to a Cameo SysML model

Note: All experiments described here were done using AskSage with CUI/ITAR safe GPT 3.5 and 4 to prove their viability in a sensitive DoD environment

Example - Generating Message Library Code

- Trained a RAG dataset on the Flex DSML by providing it the public website documentation on the language (https://docs.tangramflex.io/flex/start)
- Using a variety of message formats, provided GPT portions of the published standard (PDF), and asked it to generate a Flex DSML representation of the messages defined
- Provided the generated Flex DSML code to a C/C++ code library generator

Example - Generating Message Library Code

MESSAGE NUMBER/NAME: MT-031/ELEMENT HEARTBEAT

SIGNAL NAME/SYMBOL	LOCATION			RANGE OF VALUE	UNITS	REMARKS
	WORD	# OF BITS	SBP			
Message Time Tag (MTT)	0-1	64	31		Seconds	Double precision word. Time is in seconds since midnight 1/1/1970 (Unix EPOCH).
Message Size (MS)	2	16	31	Fixed	Decimal	Set to 4.
Message Type (MT)	2	16	15	Fixed	Octal	Set to 031.
Reserved	3	8	31			
Reserved	3	8	23			
Enclosure ID (ENCLID)	3	8	15	0 to 255	Decimal	Set to 0. 0 = Not Used
Element ID (ELEMID)	3	8	7	Fixed	Decimal	Set to 37 (Radar A).

Documentation Table

LLM

```
// Element Heartbeat message struct
message struct ElementHeartbeat {
    // Time in seconds since midnight 1/1/1970 (Unix EPOCH)
    mtt : float64; // Message Time Tag (64 bits)

    ms : uint16; // Message Size (16 bits, fixed to 4)

    mt : uint16; // Message Type (16 bits, fixed to 031 octal)

    // Reserved fields (16 bits total, 8 bits each)
    reserved1 : uint8;
    reserved2 : uint8;

    encId : uint8; // Enclosure ID (8 bits, fixed to 0)

    elemId : uint8; // Element ID (8 bits, fixed to 37 for Radar A)
}
```

DSML Code

Code Gen Tool

C++ Message Library

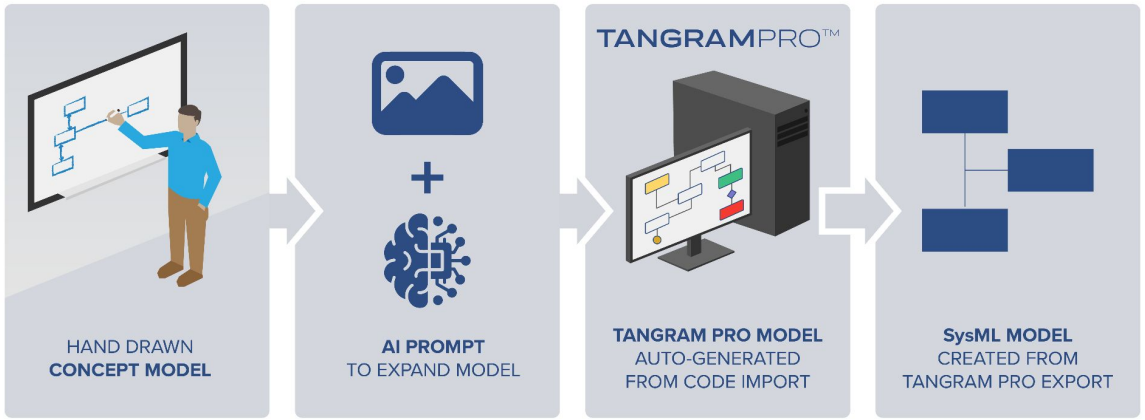
Example - Generating Message Library Code

- We measured *significant* time savings with this approach
- We gave the same challenge to 2 interns
  - Scenario: Develop a message library for STANAG-4586 (NATO standard for UAV payload and platform control) using Flex code generator
  - Neither had experience with STANAG-4586 or Flex or developing a message library
  - Both had similar academic and engineering experience
  - One intern did it without generative AI, other with generative AI
- Result: the intern with generative AI accomplished the task weeks faster

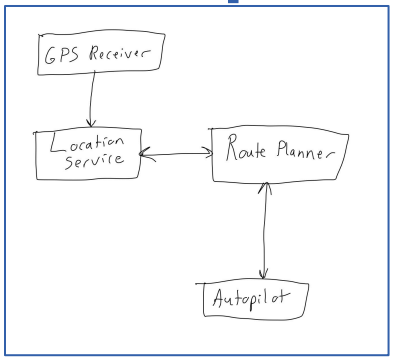
Example - Generating SysML Model from Sketch

- Trained a RAG dataset on the component modeling specification language used under the hood by Tangram Pro Designer
- Provided a hand-drawn block diagram, and prompted the LLM to generate the corresponding component model
- Synchronized the component model into a SysML internal block diagram
- Repeated the test with prompts to the LLM to make modifications to the sketch

Example - Generating SysML Model from Sketch



Example - Generating SysML Model from Sketch



Hand Drawn Diagram

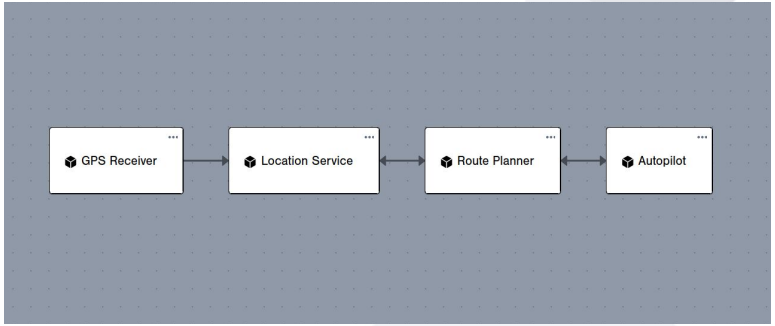
LLM

between "Route Planner" and "Autopilot."


```
{
  "name": "Autopilot Integration System",
  "description": "A system that integrates GPS data with an autopilot through a route planner with bi-directional connections.",
  "branches": [
    {
      "name": "master",
      "description": "Main integration branch for the autopilot system with bi-directional connections.",
      "head": {
        "id": "head-id-123",
        "name": "head-name-autopilot-system",
        "type": "PROJECT",
        "description": "Head of the master branch for the autopilot integration system.",
        "source_repo_url": "",
        "source_repo_revision": "",
        "properties": null,
        "subcomponents": [
          {
            "id": "gps-receiver-id",
            "name": "GPS Receiver",
            "type": "BASIC",
```

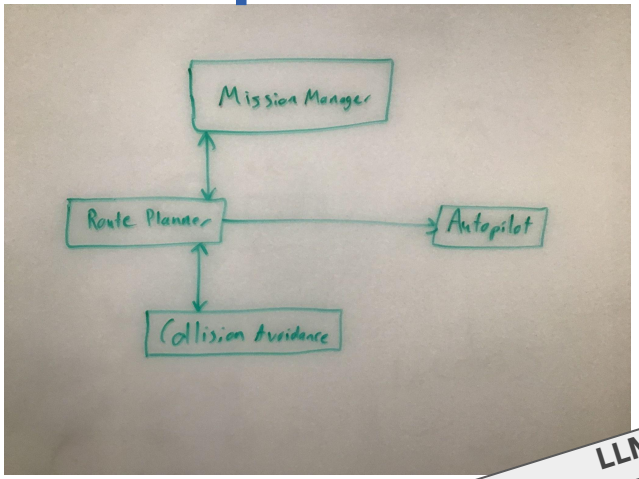

Generated DSML Code

DSML Import



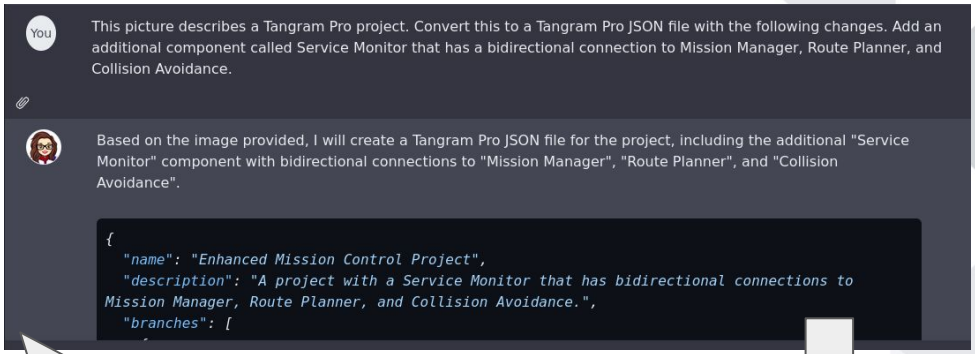
Digital System Model

Example - Generating SysML Model from Sketch



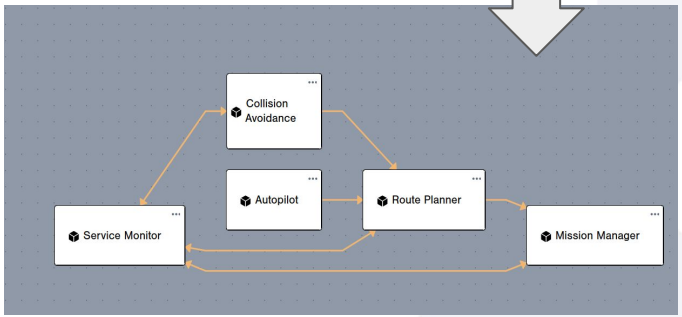
Whiteboard Diagram

LLM



Prompted changes and generated DSML code

DSML Import



Digital System Model

Example - Generating SysML Model from Sketch

- The LLM plus DSML tools were able to turn a hand drawing/whiteboard sketch into a digital model block diagram in minutes
- No straightforward path that we knew of to generate a SysML model from LLM without help of DSML/external tooling
- These results would likely hold true with voice/audio recordings as well
  - It should be possible to explain a design and have an LLM generate a digital model of the design, via an appropriate DSML

Final Thoughts for Today

Final Thoughts for Today

- DSMLs can address the gap between current generative AI technology and current systems engineering modeling practices
- When used effectively, LLMs that produce DSML output for the purposes of modeling can provide significant time savings over manual model creation
- Models still must be verified for accuracy, but verifying DSML models first can produce more accurate models in the end product format
- Let’s lean into the technology we have today to go faster!

TANGRAMFLEX®

LLM Co-pilots for Domain Specific Modeling Languages

AI4SE & SE4AI Research and Application Workshop
September 18, 2024

Matt Naveau, Chief Technology Officer
matt.naveau@tangramflex.com