

# Exploring Large Language Models for Scenario Generation in Support of C2SIM Autonomous Systems Ontology Extension Development

Paulo Barthelmess  
Hyssos Tech  
paulo.barthelmess@hyssos.com

Curtis L. Blais  
Naval Postgraduate School  
clblais@nps.edu

Keywords:

C2SIM, Autonomous Systems, ASX, Ontology, Large Language Models, LLM, GPT-4, Scenario Generation.

**ABSTRACT:** *The C2SIM Autonomous Systems sub-group has embarked on an explorative study employing GPT-4, a Large Language Model (LLM), to facilitate scenario development foundational to extending ontologies. This paper outlines the initial findings of GPT-4's application in generating context-specific scenarios, highlighting both its utility and limitations. We detail the methods adopted for directing GPT-4's output, including 0-shot learning and prompt engineering, which serve as techniques for curating scenario content in line with C2SIM requirements.*

*These methods offer a novel approach to not only summarizing existing knowledge in the literature but also in extracting embedded domain knowledge from the model, contributing to a dynamic, user-guided refinement process for scenarios. The insights from this investigation reveal the practical implications of deploying LLMs in scenario generation, thereby informing subsequent research trajectories focused on synthetic data contributions to ontology development. The paper concludes by mapping out potential avenues for future inquiry, tempered by lessons learned from the current application of LLMs in this domain.*

## 1 Introduction

In 2020, the Simulation Interoperability Standards Organization (SISO) approved the Command and Control Systems - Simulation Systems Interoperation (C2SIM) standard [1]. C2SIM standardizes information interchange across command and control (C2) systems, simulation systems, and robotic and autonomous systems (RAS) for scenario initialization, orders, and reports. The C2SIM standard consists of a core logical data model, a Standard Military Extension (SMX), and a Land Operations Extension (LOX) [2]. The SMX extends the Core C2SIM ontology by defining additional concepts and relationships to the core (as well as extending the “Code” classes with additional ontology individuals representing valid code strings for those classes) relevant to general military operations. The LOX ontology extends the combined Core + SMX ontology by adding information content particular to the land operations domain. The SISO C2SIM Product Development Group (PDG) also produced a guidance product [3] to assist early adopters of the C2SIM standard. The standard is now managed in SISO by a combined PDG/Product Support Group (PSG). As other warfighting domains employ C2SIM, the intent is to develop additional extensions to the standard to meet the new information exchange requirements.

In particular, the C2SIM PDG/PSG initiated an autonomous system extension (ASX) subgroup to more fully represent information exchange requirements for interchange of information with robotic and autonomous systems. The sub-group is currently collecting requirements, applying a scenario-based approach outlined in SISO’s Guidelines for Scenario

Development (GSD) [4]. As part of this effort, the group started an investigation into the capabilities of AI tools to facilitate crafting of scenarios. In particular, there was interest in exploring to what degree Large Language Models (LLMs) could contribute to the task.

In particular, the C2SIM PDG/PSG initiated an autonomous system extension (ASX) subgroup to more fully represent requirements for interchange of information with robotic and autonomous systems. The ASX subgroup is engaged in defining operational use cases, or scenarios, to identify the information exchange requirements necessary for the C2SIM extension. This involves reviewing several previous activities to extract concepts for information exchange and viable scenarios that will guide the effort. Amongst other activities, the group is exploring the use of Large Language Models (LLM) to assist in the creation of autonomous systems scenarios. Given the present interest in these techniques, the ASX subgroup was motivated to share this exploration and the insights gained from it with the broader SISO community.

LLMs, as described by OpenAI's GPT-4, "are advanced AI algorithms that understand, generate, and interact with human language. They are "large" due to their vast number of learning parameters and are trained on huge datasets from the internet to predict text sequences. LLMs can perform tasks like writing text, summarizing documents, answering questions, and more."<sup>1</sup>

We present the approach and methodology employed to create autonomous systems scenarios using OpenAI's GPT-4, a prominent, pioneering LLM. In summary, "prompt engineering" [5] techniques were explored in an iterative process to incrementally improve the quality of the produced scenarios. Details of the approach, the development cycle and the evaluation used are provided in Section 3.

Prompt evolution was performed mostly in the context of extraction of scenario descriptions from technical papers that discussed both hypothetical and experimental scenarios involving autonomous systems (Section 4.1). These experiments investigated the ability of the model to (radically) reformat technical information presented in a variety of formats, rendering that into a standardized format.

Once results were deemed adequate, the experiments were extended to consider the capability of the model to produce variations. Examples include prompting it to 1) change the setting used, such as from an urban environment to a tropical forest, or 2) restrict/expand the types systems employed, for example using just ground vehicles, combined ground and air, or 3) using single or multiple assets. In these instances, knowledge that is encoded in the model itself, as a representation of the collective documents ingested during LLM training is what drives the choices when producing these variations. This demonstrates the model's ability to produce synthetic scenarios that are based on an existing descriptions, but are greatly modified with information that is not contained in the papers themselves (Section 4.2). Exploitation of the encoded domain model for generation of purely synthetic scenarios – not grounded on an existing paper - is explored in the next set of experiments described in Section 4.3.

Section 5 summarizes lessons learned and discusses potential future work.

## 2 C2SIM Standard and ASX Considerations

C2SIM is an international standard produced by SISO for specifying information interchange across C2 systems, simulation systems, and robotic and autonomous systems (RAS). Figure 2.1 illustrates the C2SIM concept for information interchange across different classes of systems in a C2SIM coalition of systems. The C2SIM standard focuses on the information to be exchanged, not on the mechanism for performing the information exchange. Thus, it is transport-layer agnostic—C2SIM messages can be passed between systems by any means available, including, for example, the use of distributed simulation

---

<sup>1</sup> Created in OpenAI's Playground Chat with the prompt: "What are LLMs? TL;DR" [16]

standards such as Distributed Interactive Simulation (DIS) and High Level Architecture (HLA) for exchange of initialization data, orders, and reports across federated simulations in the C2SIM coalition.

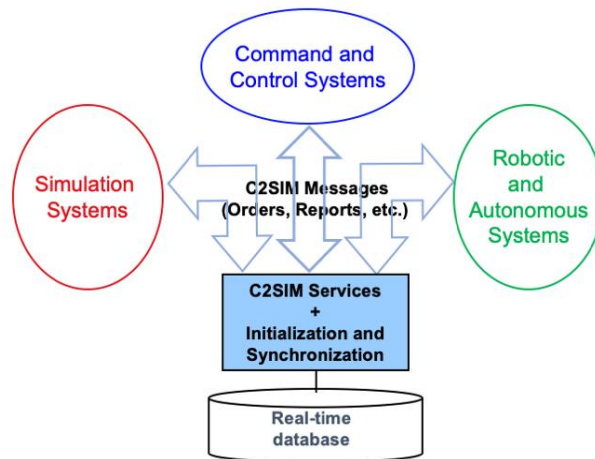


Figure 2.1: C2SIM Concept (from [1])

The C2SIM information exchange data model is specified as an ontology using the Web Ontology Language [6]. Use of an ontology permits expression of stronger semantics, offering benefits in automated reasoning, strong validation, query, and information-linking [xx]. As an ontology, C2SIM is expressed as a hierarchy of classes representing information commonly exchanged across C2 systems, simulation systems, and robotic and autonomous systems (RAS). The C2SIM standard provides a starting point for defining information interchange across these diverse systems. Development of C2SIM extensions is expected and encouraged across the community to enable information interchange in multiple warfighting domains and across numerous classes of systems (not just C2, simulation, and RAS). Example extensions include the Land Operations Extension to C2SIM [2] and a proposed extension for exchange of cyberwarfare information [7]. The C2SIM guidance product provides information to help new users develop applications that are conformant to the C2SIM standard. Additionally, there are initial efforts to create explicit integration of information from the NATO Education and Training Network Federation Object Model (NETN-FOM) with the C2SIM data model [8]. Furthermore, NATO has adopted C2SIM as Standardization Agreement 4856, Edition 0 [9] and has approved its use for integration of modeling and simulation into the NATO Federated Mission Networking program [10].

The standardized C2SIM Core and initial example extensions (SMX and LOX) have basic concepts that can support information interchange with robotic and autonomous systems (RAS). However, it is clear that fully addressing RAS information exchange with other classes of systems requires work to specify domain-specific concepts, just as is the case for the initial extensions and others that have been or are being explored (e.g., Cyberwarfare extension, Electronic Warfare extension, Maritime Operations extension, etc.). For this reason, the C2SIM PDG/PSG launched an Autonomous System Extension (ASX) subgroup to begin analysis and specification of an ASX for C2SIM. The ASX subgroup is working on description of operational use cases (i.e., *scenarios*) to identify information exchange requirements for the C2SIM extension. Several prior activities are being examined to identify concepts for information exchange and reasonable scenarios to inform the work, such as: (1) past investigations into this domain using the previous Coalition Battle Management Language (C-BML) [11] and early studies testing the C2SIM standard [12]; (2) existing RAS command and control languages, such as the Joint Architecture for Unmanned Systems (JAUS) [13] and the Autonomous Vehicle Command Language (AVCL) [14]; (3) specialized ontologies such as the Mission Execution Ontology (MEO) for ethical control of unmanned systems [15]. While examination of such activities and products continue to determine what can be learned from them, there has been curiosity about the use of large language models to assist in creating autonomous system scenarios that might further inform the work. With the current interest in such techniques, the ASX subgroup felt the SISO community would like to hear about

this exploration and what has been learned from it.

### 3 Approach to GPT4 -Supported Scenario Crafting

We exploit OpenAI's GPT-4 model [16] as an aid in the generation of autonomous systems scenarios. We examine the model's capabilities using a "0-shot" approach, with no model fine-tuning via examples (few-shots) or other customization [17]. The primary means used to shape GPT-4's responses is based on "Prompt Engineering" [5]. Prompts are natural language descriptions of the output, but carefully crafted to steer the model into producing the desired results. Results may vary, sometimes dramatically, depending on the wording, level of detail, or instructions that are provided in a prompt. Prompt refinements therefore require an iterative development approach.

#### 3.1 Development cycle

Development of prompts followed a cycle where prompts were incrementally refined to address issues detected during evaluation. Development was done within OpenAI's Playground, a simple but effective web-based environment for defining and testing prompts. The Playground interface was used to define "Assistants" containing prompt instructions. All Assistants used the "gpt-4-turbo-preview" model<sup>2</sup>.

Prompt evolution was based on the best practices listed in OpenAI's documentation [5]. Multiple prompt versions were created and tested, with details added, edited or removed incrementally to address perceived flaws in the generated output. Prompt details are presented in Sections 4.1.1 and 4.3.1.

Ideally, evaluation of prompt versions (or fine-tuned models in general) would be based on quantifiable measurements, such as the percentage of correct results produced by a version under evaluation over a set of test cases for which expected results are known. Here, there are no precise canonical scenarios that could be used as a benchmark, since there is no single way to word a scenario. Evaluation of the results was therefore based on qualitative analysis of the output produced by each version with respect to different sections of the desired standard scenario format.

Development was driven primarily by evaluating the quality of the prompts in the scenario extraction from papers task (Section 4.1), so the evaluation could be performed based on how well the model's results matched what was expressed in the papers themselves. The following questions were considered, rated in a scale from 0 to 5:

- Does the result contain all elements of the standard scenario template? Evaluates whether there are omitted (or added) elements that deviate from the desired outcome.
- Do the results reflect just the scenarios contained in the paper? Evaluates the success in extracting the scenarios from the papers' broader consideration and discussion. In many cases the scenarios are just a small part of the overall papers and may be used only for illustrative purposes, which poses difficulties for extraction.
- Is the Description a fair summary of the paper scenario? Evaluates how closely the generated summary matches the "gist" of the paper descriptions and does not contain fantasized details ("hallucinations" [18]).
- Are the generated Goals and Measurements of Performance reasonable given the context?
- Are the Steps a logical temporal progression that reflect the scenario?

Given that different results are produced in sequential runs with the same input, the evaluation took into consideration the average over sets of five runs per prompt version. The next section overviews the experiments that were conducted..

---

<sup>2</sup> OpenAI makes available a range of models, from older, less powerful ones, that are faster and cheaper to run, to more recent and advanced models. At the time of writing, "gpt-4-turbo-preview" was the most advanced (and slow) model.

## 4 Experiments

Experiments focused on three tasks: (1) extraction of scenarios from technical papers, (2) generation of synthetic variations of paper scenarios, and (3) generation of synthetic scenarios drawing from the model’s domain knowledge.

Prompts were developed primarily in the context of the first task – extraction of scenarios from papers – as this task provided better opportunities for evaluating the quality of the results based on the known content of the papers (Section 4.1.1).

The second and third tasks exploited the domain knowledge encoded in the model itself, while offering opportunities for users to control the generation process (Sections 4.2 and 4.3).

### 4.1 Extraction of scenarios from papers

We first examined LLMs capabilities as a “smart formatters”, to aid in extracting standard scenario descriptions from closely related literature. The experiments were run on a small set of initial papers contributed by the ASX group members [19], [20], [21], [22], [23], [24].

Corona & Biaginni [19] and Biaginni et al. [20] were of particular interest, because of their focus on the specific problem the sub-group is working on – extending C2SIM to include the representation of autonomous systems.

Prompts were iteratively refined to overcome perceived flaws and inconsistencies of the results (as described in Section 2). Since different responses may be returned by GPT-4 for runs with the same input, each version was evaluated considering five runs. Once results started to be considered adequate, prompt versions were used to process the remaining papers of the set, to verify their generality, and avoid the risk of overfitting the prompt, i.e., making it too specific to a particular paper.

#### 4.1.1 Prompt

Prompts can be formatted fairly freely, and there are in general no requirements for particular keywords or structure to be used. The main objective is to provide the model context enough to shape the results as intended by the prompt developer. Below we present segments of the prompt that was manually evolved, discussing the role of each piece. Notice that in practice the text is input as Instructions to the model as a single piece of text – the segments are separated here just for readability. The headers are optional and help mainly the developers identify the general instruction areas.

<b>Persona</b> You are an autonomous systems researcher describing with accuracy military scenarios contained in technical papers.
--

The persona definition sets the overall tone of the results. A “patient teacher persona, providing answers to a five-year-old”, for example, results in responses that are deemed by the model to match that particular audience. In the present case, it aligns the results with the intended military autonomous systems scenarios which are the focus of the prompt.

<b>Input</b> Extract the scenario details described in the attached file. Focus on the illustrative scenarios themselves and their goals, rather than the broader description of the paper itself.
--

OpenAI’s Playground allows for files to be attached as part of Assistant definitions, to ground the model. The instructions labeled Input direct the model to open the file and extract the information of interest as described. Otherwise, the model may take the initiative to use the information from the file or not, depending on other instructions or the users’ input. The focus on the scenarios was added to reduce the number of instances where the results summarize the paper more broadly, rather than just the scenarios they contain. That remains a problem with the current prompt – the results of some runs may

fail to focus narrowly on the scenario descriptions only. That is particularly true for papers where the scenarios represent a very small part of the discussion.

**Output Format** your description according to the following items:

- Summary narrative
- Specific goals to be achieved by the systems given the objectives
- Measures of Performance
- Scenario steps - these should not be vague descriptions, but actual steps to achieve the objectives given the context over time.

For the particular task of generating scenarios according to a standard format, the definition of the expected format of the results is crucial. One of the strengths of the model is its ability to take such simple direction and produce acceptable results. Notice that a significant amount of context is subsumed by these simple bullet points, for example, what a “summary” is, that a “narrative” style is desired, what are “goals” and how they may relate to “objectives”, and so on. The instructions added to the *Scenario Steps* strived to steer the model towards more well-defined, temporally ordered steps when possible.

The *Conceptual Model* is the most detailed section of the prompt. Earlier versions of the prompt had more generic requests for the production of a conceptual model. The current version specifies the steps in detail, per best-practices [5]. The text of the instructions between triple quotes (another best-practice recommendation to help the model identify the boundaries of related instructions) was in turn obtained by prompting GPT-4 [16] to “produce a list of steps required to extract an ontology”. This illustrates a powerful feature of the model – its ability to follow high-level procedures described in the prompts. This element remained the most inconsistent throughout the experiments and requires additional work - see Section 5 for a more detailed discussion of future work.

**Conceptual Model** Use the instructions within triple quotes (""") below to extract an ontology from this scenario. List the results in a section labeled "Conceptual Model"

""Domain and Scope Identification: Define the ontology's domain and scope within the military scenario context.

Concept Identification: List main concepts present in the scenarios.

Hierarchy Definition: Organize these concepts into a hierarchical structure.

Relationship Identification: Detail relationships between concepts, specifying types.

Property Identification: Note significant properties of each concept.

Constraints/Rules Identification: Enumerate any rules or constraints guiding concept relationships or properties.""

#### 4.1.2 Example

For illustrative purposes, we present the result of a single run, extracting a scenario from Corona & Biaginni [19], the primary benchmark used during prompt evolution. Additional results can be found in the ASX group GitHub working repository [25].

#### GPT-4:

The document contains a detailed description of an operational scenario aimed at demonstrating and evaluating the capabilities of unmanned autonomous systems (UxS) in military operations, particularly in an urban setting. The scenario is developed to meet specific research and operational goals for the integration of UxS within command and control (C2) structures and to explore their effectiveness in complex environments. Here's the analysis based on the requirements:

#### Summary Narrative:

---

[Type text]



The scenario revolves around a mission to protect troops and civilian populations against hostile unmanned aerial systems (UxS) in an urban environment. A team of unmanned ground vehicles (UGV) escorts a human platoon through a city while unmanned aerial vehicles (UAV) conduct reconnaissance to identify threats. The UAVs and UGVs collaborate, employing both non-kinetic and kinetic countermeasures against identified threats in a tiered defense system. The scenario is designed to test and demonstrate various aspects of UxS integration, including autonomous decision-making at different levels of autonomy (LoAs), the effectiveness of mixed human-robot teams, and the operational utility of UxS in urban combat situations.

**Specific Goals:**

1. Detection and identification of enemy robotic units using UAVs and sensors.
2. Augmentation of situational awareness with external decision-making tools.
3. Defense against enemy UxS using friendly UxS.
4. Employment of UxS in urban settings, covering both land and air domains.
5. Exploration of human-robot interaction across different LoAs.

**Measures of Performance:**

- Efficiency in detecting and identifying enemy UxS.
- The effectiveness of mixed human-robot teams in urban reconnaissance and combat scenarios.
- Level of situational awareness achieved.
- The effectiveness of countermeasures deployed against enemy UxS.
- The degree of autonomy demonstrated by UxS and its impact on mission success.

**Detailed Scenario Steps:**

1. A Command Post receives mission orders and dispatches UAV and UGV teams with optimized routes.
2. UAVs conduct reconnaissance to identify threats while UGVs escort a human platoon.
3. UxS generate status reports and, upon detecting hostile units, produce observation reports suggesting further actions while awaiting human confirmation.
4. Upon receiving orders, UAVs exit the area via designated routes, and UGVs escort the human platoon out of the danger zone, employing jamming or firing based on the proximity of the threat.

**Conceptual Model:**

- **Domain and Scope Identification:** Military operations involving mixed teams of humans and autonomous systems in urban environments.
- **Concept Identification:** Command Post, UAV Swarm, UGV Team, Human Platoon, Hostile UxS, Threat Detection, Situational Awareness, Countermeasures, Levels of Autonomy (LoA).
- **Hierarchy Definition:**
  - Actors: Command Post, UAV Swarm, UGV Team, Human Platoon.
  - Actions: Reconnaissance, Escort, Jamming, Firing.
  - Objects: Hostile UxS.
- **Relationship Identification:**
  - UAV Swarm performs reconnaissance and reports threats.
  - UGV Team escorts Human Platoon and takes defensive actions based on threat proximity.
  - Command Post coordinates overall mission execution.
- **Property Identification:** Level of Autonomy for UGV Team and UAV Swarm, Threat Detection capability, Countermeasure effectiveness.
- **Constraints/Rules Identification:**
  - UAVs must wait for human confirmation before taking suggested actions.
  - Different countermeasures (non-kinetic to kinetic) are deployed based on the threat's proximity.

This comprehensive analysis highlights the intricate coordination required between different UxS and human operators in a complex urban military scenario, showcasing the operational potential and challenges of integrating autonomous systems into traditional military structures.

---

## 4.2 Generation of synthetic scenario variations

One of the most powerful capabilities of LLMs is their ability to reshape results based on further guidance, which is then incorporated into an ongoing context. Given a result, one can for example direct the model to “make it shorter”, “make this

sound more friendly (or more professional)”, or “write this as a limerick”<sup>3</sup>. Here, we were interested in exploring how this can be used to produce variations of the scenarios extracted from papers, exploring alternatives and multiplying the number of collected scenarios with the model’s aid.

We examined variations via prompts provided to GPT-4 [16] immediately after the production of successful paper summary extractions. These prompts included changes to : (1) the types of autonomous systems employed, for example, restricting a scenario to use just UGVs (“Change this to use just UAVs”), or UAVs (“Use just UAVs”), (2) the number systems (“Rewrite this using a single UAV”, or “Use one UGV and a swarm of UAVs”), (3) the settings (“Change this to a tropical forest setting”).

#### 4.2.1 Variations Examples

The following examples show excerpts of the results produced when the corresponding prompts (**User**) were entered in the context of Corona & Biaginni [19] scenario extraction shown in Section 4.1.2. See [25] for extended versions of the variations.

---

##### User

*Change this to a tropical forest setting*

---

##### GPT-4:

Adapting the described operational military scenario from an urban to a tropical forest setting involves significant adjustments due to the vastly different environmental dynamics and operational constraints. Nonetheless, the foundational strategy, specific goals, measures of performance, and detailed scenario steps can be altered to suit the new context while still retaining the core objectives related to unmanned autonomous systems (UxS) capabilities. Here's how the analysis would be adapted:

##### Summary Narrative:

The revised scenario takes place in a dense tropical forest, aiming to protect troops and local populations against hostile unmanned aerial systems (UxS) in this challenging environment. A team of unmanned ground vehicles (UGV), modified for rugged terrain, escorts a human platoon through the forest paths, while unmanned aerial vehicles (UAV) conduct overhead reconnaissance looking for threats hidden under the canopy. The thick vegetation introduces new challenges for both detection and communication. The UAVs and UGVs collaborate to employ non-kinetic and kinetic countermeasures against identified threats in a layered defense system, testing the UxS capabilities in navigating and operating in dense forests.

##### Specific Goals:

1. Adapted detection and identification of enemy robotic units using UAVs fitted with specialized sensors for forest environments.
2. Augmentation of situational awareness in a dense forest, leveraging external decision-making tools.
3. Deployment of UxS for effective defense against enemy UxS in forest terrain.
4. Exploration of UxS employment in land domain within a tropical forest, dealing with vegetation and terrain challenges.
5. Investigation of human-robot interaction across different Levels of Autonomy (LoAs) in a non-urban setting.

##### Measures of Performance:

- The effectiveness in detecting and identifying enemy UxS through dense foliage.
- The efficiency of mixed human-robot teams in forest reconnaissance and threat neutralization.
- The level of situational awareness achieved under forest canopy coverage.
- The success of countermeasures (non-kinetic and kinetic) in neutralizing threats in a forest environment.

---

<sup>3</sup> This paragraph, rendered by OpenAI [16] as limerick: “LLMs boast power quite fantastic, To mold results with tweaks elastic. "Shorter, friendlier," you can direct, Or "more professional," to perfect, Even ask for a limerick, oh so classic!



- The impact of autonomy levels on operational success in forest terrains.

#### **Detailed Scenario Steps:**

- A Command Post, possibly mobile or at the forest edge, receives and disseminates mission orders, deploying UAV and UGV teams with routes optimized for forest terrain.
- UAVs, using advanced sensor technology, perform surveillance from above the canopy to detect threats, while UGVs navigate forest paths to escort a human platoon.
- Upon threat detection, UAVs and UGVs generate and share detailed observation reports, suggesting actions but awaiting confirmation from human operators.
- Upon receiving confirmation, UAVs exit the threat area using pre-determined routes, while UGVs employ countermeasures tailored to forest engagement - using jamming devices or kinetic actions based on threat proximity and terrain constraints.

---

Other variations can be created for example by requesting the model to “Generate 5 very distinct variations of the scenario. There are no restrictions on the type and number of autonomous systems employed.” Results include context-appropriate adaptations for “Arctic Reconnaissance Mission”, using sensors to detect enemy structures buried under the ice, “Desert Warfare and Convoy Protection”, “Urban Disaster Response”, “Jungle Drug Lab Elimination”, and “Offshore Energy Infrastructure Protection”, the latter employing surface and subsurface assets (see [25]).

Independently of the value of the scenarios generated by the model in terms of their contribution to the development of an autonomous system ontology, it is important to consider where the new information that is being (coherently) applied is coming from. LLMs, due to the very nature of their training, which includes very large numbers of different sources, encode a certain degree of domain knowledge over a wide range of subjects, which we see surface here.

In the next section, synthetic scenarios created in absence of a baseline paper are explored..

### **4.3 Mining the model’s domain knowledge**

Domain and common sense knowledge embedded in LLMs have been exploited in a few areas such as robot planning [26] [27], and has been examined in terms of embedded factual geographic knowledge [28].

Given the lack of curation of the sources used to train models, and the opaque nature of the sources used, there are risks associated with using models as sources of knowledge. Bias, erroneous information and unknown gaps in training make it risky to rely on such types of information in models that are not fine-tuned and grounded on a volume of trusted authoritative documents. The problem is made more acute by the artful way in which models present misinformation, or erroneously combined pieces of information, which makes it harder to detect flaws.

Our primary interest though is in providing tools for expert users to facilitate the production of scenarios and exploration of variations in a way that makes the best of their expertise and time. As such, there is an assumption that the essential aspects of the scenario definition are a result of the experts’ guidance, and that outputs are examined for quality. The main Assistant is a *Scenario Synthesizer*, which takes Objectives and a Context as user input, and produces scenarios formatted according to the ASX group’s standard. Additional Assistants can be used to produce (1) lists of *Areas and sub-areas* where autonomous systems are employed, and (2) *Objectives given an Area*. The expectation is that expert users may employ the main *Scenario Synthesizer*, while the two other ones may be leveraged to provide potentially different perspectives to experts, or serve as a source of information for users with less expertise.

#### **4.3.1 Prompt**

The prompt that was evolved to support the extraction of scenarios from papers (Section 4.1.1) served as the new prompt’s foundation. Changes were made to the Persona and Input segments of the prompt only, with the output instructions remaining the same.

**Persona** You are an autonomous systems military expert, describing with accuracy scenarios employing autonomous systems to achieve the objectives the user provides.

The persona was adapted to align it to an expert role in the domain, capable of defining scenarios compatible with user-provided high-level input.

**Input** The list of Objectives provided by the user are prefixed with the key work "Objectives:". The user may also provide a context, prefixed with the keyword "Context:". If not context is provided by the user, propose one yourself.

Rather than take input from a file, as was the case previously, here the user provides the input as a prompt when running the assistant. The specification of the expected pieces that the user provides is meant to help parse the user input, even though considerable flexibility can be observed with respect to what can be actually provided in the prompts.

#### 4.3.2 Example

To provide means to directly compare the synthesized scenario with one extracted from a paper, we extracted the Objectives and Context from Corona & Biaginni [19], and provided that as the (**User**) input prompt<sup>4</sup>.

---

##### User

###### Objectives:

- Detection and Identification of enemy robotic units utilizing UAxS and sensors;
- Situational Awareness augmented with external decision-making support tool;
- Experiment about defense against UAxS using UAxS;
- UAxS employed in urban environment both in land and air domain

*Context: Mission for protecting the troops and populations against hostile UAxS in modern urban environment. A team of unmanned ground vehicles (UGV) escorts a human platoon in a city while a swarm of unmanned air vehicles (UAV) performs a reconnaissance of the area searching for threats. As soon as hostile air C2SIM Operationalization Extended to Autonomous Systems drones show up, UAVs generate reports on enemy activity for UGVs which activate a two levels defensive system based on a safety bubble where, according to the proximity of the threat, the countermeasures increase from non-kinetic (jamming or capturing) to kinetic (shooting).*

---

##### GPT-4:

###### Summary Narrative

The proposed scenario involves an orchestrated mission to safeguard troops and civilian populations in a modern urban setting from hostile unmanned aerial systems (UAxS). It incorporates a blend of unmanned ground vehicles (UGV) and unmanned aerial vehicles (UAV) to form a comprehensive defense strategy. The UGVs serve as escorts to a human platoon, offering ground support and establishing a defensive perimeter. Concurrently, UAVs are deployed for aerial reconnaissance, tasked with identifying and tracking enemy robotic units. Upon detection of threats, these UAVs relay pertinent information back to the UGVs, triggering a tiered defense protocol that scales from non-kinetic to kinetic countermeasures based on the threat's proximity. This strategy not only aims to neutralize threats but also to collect data for an external decision-making support tool, enhancing future situational awareness and response strategies.

###### Specific Goals

1. **Detection and Identification:** UAVs will employ advanced sensors to detect, track, and identify enemy UAxS, focusing on real-time threat assessment.

---

<sup>4</sup> The generated Scenario Steps and Conceptual Model are not included for space considerations. See [25] for details.

2. **Situational Awareness:** Data gathered by UAVs will be integrated into an external decision-support system, providing comprehensive situational awareness to command units.
3. **Defense Experimentation:** Implementation of a dual-tier defense mechanism against hostile UAxs, testing both non-kinetic (e.g., jamming, capturing) and kinetic (e.g., direct engagement) countermeasures.
4. **Urban Operation:** Successfully operate the UAV and UGV systems within a complex urban environment, ensuring minimal collateral damage and civilian safety.

#### Measures of Performance

- **Enemy UAxs Detection Rate:** Percentage of hostile units successfully detected by UAVs.
- **Identification Accuracy:** Accuracy of identification systems in classifying threats.
- **Response Time:** Time taken from threat detection to activation of appropriate countermeasures.
- **Incident Resolution Rate:** Percentage of hostile UAxs neutralized or rendered inoperative.
- **Collateral Damage Assessment:** Quantitative measurement of unintended damage to urban infrastructure or harm to civilians.
- **System Endurance:** Duration UAxs can operate without refueling or maintenance.

---

## 5 Discussion and Future Work

LLMs' strong linguistic capabilities resulted in consistently high-quality, concise, expressive descriptions. Reasoning capabilities proved useful (and usable) in the generation of Steps, Goals, and Measures of Performance. A considerable amount of inference is applied to interpret what "summary", "goals given objectives", "measures of performance", and temporally ordered "steps" mean in a particular context. A complex pipeline of Natural Language tools, out of reach for most users, would be required to obtain similar results. It is likely that a conventional approach would be less generic, narrowly focusing on a particular domain.

The ability to bring varying degrees of additional model-encoded domain knowledge to bear either via prompt design or in a dialog fashion, where changes and adaptations can be requested within an ongoing evolving context is also something hard to replicate with other technologies. This ability provides for a seamless experience to users wanting to adapt or refine a scenario, or to explore different alternatives.

Natural language instructions provide on the one hand fluid, accessible means to shape results, including the definition of procedures/reasoning that should be followed to produce the result. On the other hand, small changes to the instructions can have great impact in hard to predict ways. The prompt crafting process therefore requires a considerable amount of testing to evaluate the impact of each change. Other than the generic best practices [5], which proved effective, there are currently no precise means to engineer prompts with higher degrees of confidence on how changes affect results other than by trial and error.

Quality of the results remains at this point somewhat uneven, particularly in the more challenging task of extracting scenarios from papers, with occasional production of results that deviate significantly from the expected. Future work will investigate prompt improvements and the use of additional fine-tuning techniques [17] to overcome that. The extraction of a conceptual model, which is prone to produce different results at every run, will likely require implementing ontology-specific techniques such as the ones described by Giglou et al. [29], perhaps in combination with traditional ontology learning approaches.

The usefulness of purely synthetic scenarios has not been fully determined. Scenarios are created by subject matter experts as vessels for aspects of importance, drawn from their deep insights into the problem space. Although LLMs can generate many synthetic scenarios, it is uncertain to what extent they contribute to the capturing of an ontology. The approach we aim to further explore involves experts utilizing the technology to facilitate the creation of relevant scenarios, including easy generation of variations that may help them illustrate key points. Also of interest is the whether easy access to a significant amount of domain data encoded in the model itself can benefit less expert users, as they explore and learn. Future work includes refining the set of assistants based on feedback from these users to better support their needs.

## 6 References

- [1] Simulation Interoperability Standards Organization, "Standard for Command and Control Systems – Simulation Systems Interoperation (C2SIM)," SISO-STD-019-2020, 2020.
- [2] Simulation Interoperability Standards Organization, "Standard Land Operations Extension to Command and Control Systems – Simulation Systems Interoperation (C2SIM)," SISO-STD-020-2020, 2020.
- [3] Simulation Interoperability Standards Organization, "Guide for Command and Control Systems – Simulation Systems Interoperation (C2SIM)," SISO-GUIDE-010-2020.
- [4] Simulation Interoperability Standards Organization, "Guideline on Scenario Development for Simulation Environments," SISO-GUIDE-006-2018, 2018.
- [5] OpenAI, "Prompt engineering," 2024. [Online]. Available: <https://platform.openai.com/docs/guides/prompt-engineering>. [Accessed February 2024].
- [6] World Wide Web Consortium (W3C):, "OWL 2 Web Ontology Language Document Overview (Second Edition), W3C Recommendation 11," December 2012. [Online]. Available: <https://www.w3.org/TR/owl2-overview>.
- [7] C. L. Blais, "An Update on Efforts to Extend the Command and Control Systems – Simulation Systems Interoperation (C2SIM) Standard for Exchanging Cybersecurity Information," in *Simulation Innovation Workshop - SIW*, 2022.
- [8] C. L. Blais, "Reconciling the Command and Control Systems – Simulation Systems Interoperation (C2SIM) Standard with the NATO Education and Training Network (NETN) Federation Object Model," in *Simulation Innovation Workshop - SIW*, 2022.
- [9] NATO, "Standardization Agreement for Command and Control Systems – Simulation Systems Interoperation," STANAG 4856, Edition 0, 2023.
- [10] K. Galvin, J. M. Pullen, H. Jense, C. L. Blais, T. van den Berg and M. Dechand, "Advancing Modelling and Simulation in NATO Federated Mission Networking," in *Proceedings of the Fall NATO Modelling and Simulation Symposium*, Monterey, CA, 2023.
- [11] Simulation Interoperability Standards Organization, "Standard for Coalition Battle Management Language (C-BML)," SISO-STD-011-2014, 2014.
- [12] C. L. Blais, "A Robotics and Autonomous System Use Case to Guide Specification of the Command and Control Systems - Simulation Systems Interoperation (C2SIM) Standard," in *Simulation Innovation Workshop - SIW*, 2019.
- [13] Joint Architecture for Unmanned Systems, "Reference Architecture Specification," Volume II, Part 1, Architecture Framework, Version 3.3, 2007.
- [14] D. Davis, C. L. Blais and D. Brutzman, "Autonomous Vehicle Control Language for Simulated and Real Robotic

Forces," in *Simulation Interoperability Workshop - SIW*, 2006.

- [15] D. Brutzman, C. L. Blais, H. Wu, Andersen and R. Markeloff, "Autonomy Compliance with Doctrine and Ethics by Using Ontological Frameworks," in *Trolley Crash: Approaching Key Metrics for Ethical AI Practitioners, Researchers, and Policy Makers*, Academic Press, 2024.
- [16] OpenAI, "Large Language Model - gpt-4-turbo-preview," 2024. [Online]. Available: <https://platform.openai.com/playground>. [Accessed February 2024].
- [17] OpenAI, "Fine-tuning," 2024. [Online]. Available: <https://platform.openai.com/docs/guides/fine-tuning> . [Accessed February 2024].
- [18] Y. Zhang, Y. Li, L. Cui, D. Cai, L. Liu, T. Fu, X. Huang, E. Zhao, Y. Zhang, Y. Chen, L. Wang, A. T. Luu, W. Bi, F. Shi and S. Shi, "Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., Huang, X., Zhao, E., Zhang, Y., Chen, Y., Wang, L., Luu, A.T., Bi, W., Shi, F., & Shi, S. (2023). Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. ArXiv, abs/2309.01219," Zhang, Y., Li, Y., Cui, L., Cai, D., Liu, L., Fu, T., Huang, X., Zhao, E., Zhang, Y., Chen, Y., Wang, L., Luu, A.T., Bi, W., Shi, F., & Shi, S. (2023). Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. ArXiv, abs/2309.01219, 2023.
- [19] F. Corona and M. Biaginni, "C2SIM Operationalization Extended to Autonomous Systems. Modelling and Simulation for Autonomous Systems," in *Modeling & Simulation for Autonomous Systems Conference, MESAS*, 2019.
- [20] M. Biagini, F. Corona, F. Innocenti and S. Marcovaldi, "Requirements and Example for a C2SIM Extension to Unmanned Autonomous Systems (UAXS)," NATO M&S COE Annual Review Vol. 2, Chapter 8, p93, 2018.
- [21] D. Brutzman and C. Fitzpatrick, "Creating Virtual Environments for Evaluating Human-Machine Teaming," Naval Postgraduate School Technical Report NPS-MV-20-001, 2020.
- [22] M. Langerwisch, T. Wittmann, S. Thamke, T. Remmersmann, A. Tiderko and N. Wagner, "Heterogeneous teams of unmanned ground and aerial robots for reconnaissance and surveillance - A field experiment," in *IEEE International Symposium on Safety, Security, and Rescue Robotics, SSR*, 2013.
- [23] T. Remmersmann, U. Schade, K. Rein and A. Tiderko, "BML for Communicating with Multi-Robot Systems," in *Simulation Innovation Workshop - SIW*, 2015.
- [24] T. Remmersmann, I. Trautwein, U. Schade, B. Brüggemann, C. Lassen, M. Westhoven and M. Wolski, "Towards Duty – BML Communication Enables a Multi-Robot System Supporting an Infantry Platoon," in *Simulation Innovation Workshop - SIW*, 2016.
- [25] C2SIM ASX Group, "C2SIM Autonomous Systems (ASX) GitHub Repository," 2024. [Online]. Available: <https://github.com/OpenC2SIM/C2SIMArtifacts/tree/main/Subgroups/ASX>.
- [26] K. Rana, J. Haviland, S. Garg and J. Abou-Chakra, "SayPlan: Grounding Large Language Models using 3D Scene Graphs for Scalable Task Planning," in *7th Annual Conference on Robot Learning - CoRL*, 2023.
- [27] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Trembley, D. Fox, S. Thomason and A. Garg, "PROGPROMPT: program generation for situated robot task planning using large language models," *Autonomous*

*Robot*, vol. 47, p. 999–1012, 2023.

- [28] J. Roberts, T. Luddecke, S. Das, K. Han and S. Albanie, "GPT4GEO: How a Language Model Sees the World's Geography," 2023. [Online]. Available: <https://arxiv.org/abs/2306.00020>.
- [29] H. Babei Giglou, J. D'Souza and S. Auer, "LLMs4OL: Large Language Models for Ontology Learning," in *ISWC*, 2023.
- [30] D. Truhn, J. Reis-Filho and J. Kather, "Large language models should be used as scientific reasoning engines, not knowledge databases," *Nat Med*, vol. 29, p. 2983–2984., 2023.
- [31] Microsoft, "Azure AI Studio," 2024. [Online]. Available: <https://ai.azure.com/>.
- [32] J. Nagle, N. DeLong and T. Trainor, "Initial Set of Use Cases for High-Fidelity Synthetic Environment / Virtual Autonomous Navigation Environment Development Tied to Unmanned Ground Vehicle Capability Gaps," United States Military Academy Operations Research Center of Excellence Technical Report #DSE-TR-0912, 2008.

## Author Biographies

**PAULO BARTHELMMESS** is a Principal and Co-founder at Hyssos Tech, a company specializing in Natural Language human-machine interfaces for planning. His expertise spans planning solutions, AI, and multimodal user interfaces, with contributions both in industry and academia. Dr. Barthelmess has an M.Sc. in Computer Science from the Universidade Estadual de Campinas, and a Ph.D. in Computer Science from the University of Colorado at Boulder.

**CURTIS BLAIS** is a member of the Naval Postgraduate School research faculty in the Modeling, Virtual Environments, and Simulation (MOVES) Institute. He has over 49 years of experience in all aspects of defense modeling and simulation, from specification through design, development, testing, and application. Dr. Blais is active in the Simulation Interoperability Standards Organization (SISO), with particular interest and involvement in development of standards for information interchange across broad classes of systems, to include command and control systems, simulation systems, and robotic and autonomous systems. Other research interests include modeling of human behavior, particularly in contrast to modeling of robotic and autonomous system behavior, distributed simulation system architectures, and design and application of ontologies and knowledge graphs. Dr. Blais has a B.S. and M.S. in Mathematics from the University of Notre Dame and a Ph.D. in MOVES from the Naval Postgraduate School.