Supporting Pragmatic Interoperability: An LLM Based Process to Analyze Distributed Intentionality (i*) Models

Roberto de Cerqueira Figueiredo
Institute of Computing.
Federal University of Bahia
City, Country
roberedo@ufba.br

Julio Cesar Sampaio do Prado Leite
Institute of Computing.
Federal University of Bahia
Salvador, Brazil
julioleite@ufba.br

Célia Ghedini Ralha
Institute of Computing.
University of Brasilia
Salvador, Brazil
ghedini@unb.br

Rita Suzana Pitangueira Maciel
Institute of Computing.
Federal University of Bahia
Salvador, Brazil
rita.suzana@ufba.br

Daniela Barreiro Claro
Institute of Computing.
Federal University of Bahia
Salvador, Brazil
dclaro@ufba.br

***Abstract*—Given the advent of large language models (LLM), automatic goal-based model analysis in goal-oriented requirements engineering is a new opportunity. A well-known problem within systems collaboration is the interoperability of its components. Pragmatic interoperability is more challenging than other levels (e.g., syntactic, semantic) since it depends on usage. Automatic detection of variation points in goal-based models and variant analysis are vital to improving pragmatic interoperability between components since they deal with different uses. We propose an integrative process using a distributed intentionality modeling language (i*) strategic rationale (SR) goal model with an LLM to detect independent variation points and analyze which variant is desirable to improve systems' pragmatic interoperability. The automatic analysis of the LLM is experimented with using image classification contexts to detect risk-situation objects. The detected LLM variation points in i* SR models are evaluated through a controlled experiment, calculating precision, recall, and F-measure. The results present an F-measure of 0.3, indicating that the proposed process is promising in improving pragmatic interoperability.**

***Index Terms*—Goal-oriented Requirements engineering, intentional modeling, pragmatic interoperability**

I. Introduction

Pragmatic interoperability refers to the mutual understanding of how data exchanged between systems is used, going beyond the syntactic (format) and semantic (meaning) levels of interoperability. It concerns ensuring that systems have the same understanding of the intended effects of exchanged messages in a specific context [1]. It is critical to the collaboration between systems, ensuring that the exchanged data is understood and acted on consistently in a given context [2].

Pragmatic interoperability can be defined as:

$$I_i(D_{ij}) = I_j(D_{ij}) \tag{1}$$

wherein I_i and I_j are system interpretation functions S_i and S_j', respectively, and D_{ij} denotes the exchanged data. This ensures that both systems derive equal actionable meaning from a message. The message must be interpreted in the correct context, aligned with the intention of the sender, actionable, and useful for decision making.

Intentional models can help interpret the intended effect of the message. Strategic Rationale (SR) models are intentional models that are essential for capturing and visualizing agents' intentions.

To achieve collaboration in dynamic systems, automatic analysis in SR models of the message's intended effect are desirable. For this, detecting variation points and analyzing variants are necessary. Variation points are places in design artifacts where a specific decision has been narrowed to several options but the option to be chosen for a particular system has been left open [3]. In our case, variation points are the goals (ends) which are refined into tasks (means), such that decisions could vary by analyzing variants' impact. Variation points are the model's place where decisions or actions could vary by analyzing variants' impact.

This work proposes a novel process for detecting variation points and analyzing their variants in the SR model with a LLM. We provide the LLM with a JSON file of an SR model and the text(background information) of different scientific documents that are used to obtain concepts and knowledge needed for the task at hand. A controlled experiment analyzed the LLM performance. We evaluated how using scientific documents on different prompts can impact LLM performance by detecting independent variation points. Independent variation points are variation points that are not refined further. Next, we used the detected variation points to identify their variants and select which variants are more appropriate for use in

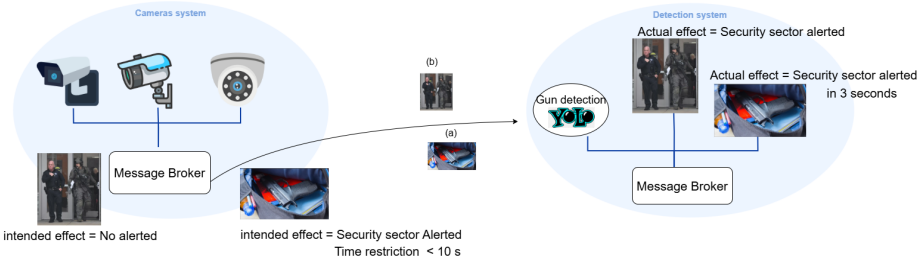

Fig. 1. The camera system produces frames and stores them in its message broker. The detection system consumes the frames from the camera system's message broker, interprets the context of each scene, and decides whether it is an alert case. Two situations can occur when data is exchanged between these systems: (a) Collaboration with pragmatic interoperability; (b) Collaboration without pragmatic interoperability.

image contexts. It contributes to pragmatic interoperability because we evaluate the system capability before systems collaboration. The intended effects can be interpreted with high probability. Besides, it ensures alignment by identifying variation points and reconciling intended and actual effects during runtime.

Through our approach, we seek to answer the following research question: How does evaluating intentional models automatically improve pragmatic interoperability?

We organized this paper as follows: Section 2 describes the background, Section 3 describes related work, Section 4 details our research design choices, Section 5 shows the experiment, Section 6 defines the variants analysis and system selection, Section 7 evaluates the results, and Section 8 summarizes the contributions and describe possible future works.

II. Background

In the following subsections, we will present an example of pragmatic interoperability, system's capability modeling through goal model and variation points and variants.

A. Pragmatic interoperability problem example

We provide an example of pragmatic interoperability, stressing that a correct interpretation of intention depends on the provided context. As illustrated in Figure 1, two systems collaborate to alert the security sector about suspicious detections in security camera images. The camera system provides the images, whereas the gun detection system provides the interpretation of the image based on its context.

In the case of (b) image, two armed officers in front of a building, the detection system alerts the security sector. However, this is not a threat, so there was a misinterpretation of the intention of the situation, which was not to provide an alert. This causes a pragmatic interoperability problem, as understanding the intended effect does not align with the interpreted effect.

In image (a), a gun within a backpack inside a car, the detection system alerts the security sector. In this case, it is a threat. As such, the intention is to alert the security sector within a time window of a maximum of 10 seconds. When the detection system interprets the image and its context, it concludes that alerting the security sector is necessary. The detection system issues the alert in 3 seconds. In this case pragmatic interoperability is achieved, the effect produced matches the intention.

B. Detection systems' capability modeling

This section shows the i* strategic rationale model [16] from the systems presented in the scenario of Figure 1. An SR model is defined by an actor (circle) that sets a boundary for its goals (ovals), softgoals (curved cloud-like shapes), tasks (hexagons), and resources (rectangles), which are linked either by means-ends relationships (solid arrows) or task decomposition (crossed lines) or contributions (arrows with a text label). Different actors are linked by dependency links (D symbol indicating direction).

Previous work [20] has used i* to model pragmatic interoperability. We adapt the SR model. In this new SR model version, as illustrated in Figure 2, a second RCNN [4] detection system is incorporated in addition to the Yolo [5] detection system. This added system is a candidate for collaboration with the camera system. The detection systems' capability to interpret scenes is the focus of this modeling. SR modeled each detector's ability to detect objects in images based on the accuracy of image interpretation and the time used in interpretation. It can aid in selecting the most suitable detector to identify objects in images, thereby ensuring the correct interpretation of the camera system's intended effect. So, how can we explore using the SR model to identify the best choice between detection models to improve pragmatic interoperability?

To answer this question, initially, we must understand how the capability to interpret scenes is present in the SR model. The SR model can interpret scenes through the YOLO and RCNN agents. Both have the task of interpreting a normal scene and a scene with obstruction. These tasks, when performed, contribute positively or negatively to the accuracy and response time soft goals. These contributions are represented by relationships called contributing links. A Help Contribution Link is used when one element's existence supports or facilitates the occurrence of another. It makes a positive contribution to a system. A Hurt Contribution Link means that one element interferes, blocks, or undermines another from achieving. It reflects a negative contribution.

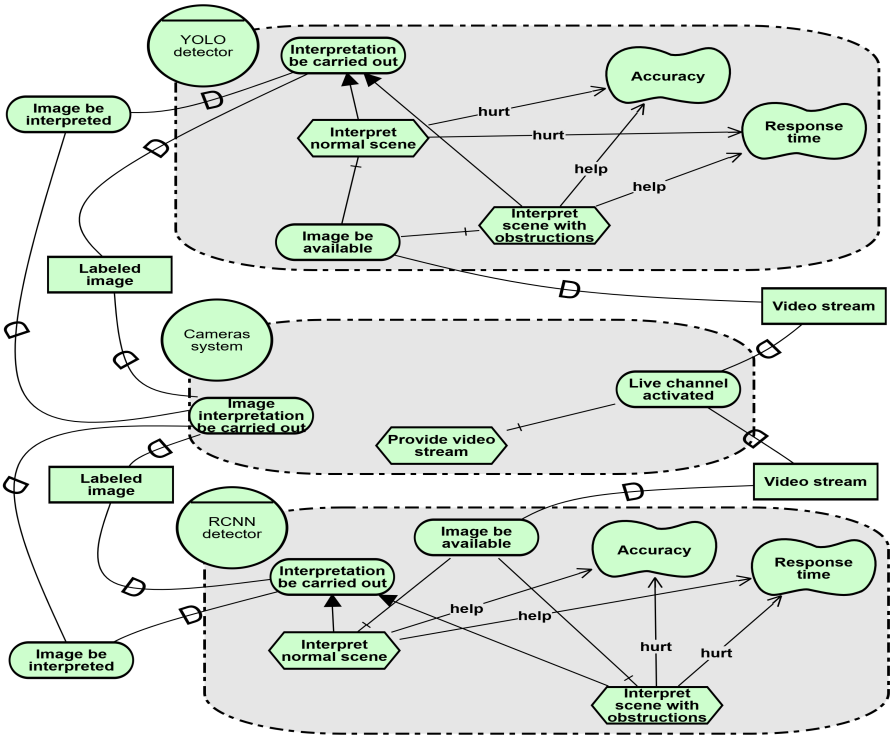

Fig. 2. Strategic Rationale goal model to detect objects in images.

C. Capabilities instances of detection models.

Detectors have capabilities that make them different. Identifying the instances of capability helps identify the differences. These instances are related to the concept of variability. Variability refers to the capability of a software product family to offer different product configurations [9]. The variability is defined by the introduction of variation points. A variation point defines a decision point with its possible choices (functions or qualities). The available functions and qualities for a variation point are called variants. When specific variants are chosen, it leads to different product outcomes [9].

A variation point in a goal model refers to a decision-making point in the design process where we choose alternative functionalities or behaviors. They represent alternative paths or solutions that fulfill higher-level stakeholder goals. In i* [16], the means-end relationship (solid arrows) models these alternatives; each alternative is a means (task) to achieve a goal. A task may be decomposed by lower-level goals through the task decomposition relationship (crossed lines). This modeling capability provides a way of describing goal refinement by different abstraction levels. Pistar 1.0, based on [15], implements the means-end relationship complaining with the original i*. As illustrated in Figure 3, each arrow represents a means-end relationship, and that two or more means-ends generate a variation point VP. If another VP is not subordinate to this VP, the VP is considered an independent variation point. So, an independent variation point (IVP) means that there will be no further goal refinement from this variation point.

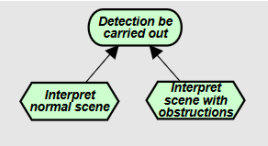

Fig. 3. Independent Variation point present in the YOLO detection and RCNN detection systems

Each means-end arrow represents a variant. Figure 4 shows the variants of the detection models, starting with a task and ending with goals and soft goals. Figure 4 illustrates two possible variants for each detected independent variation point in this example.

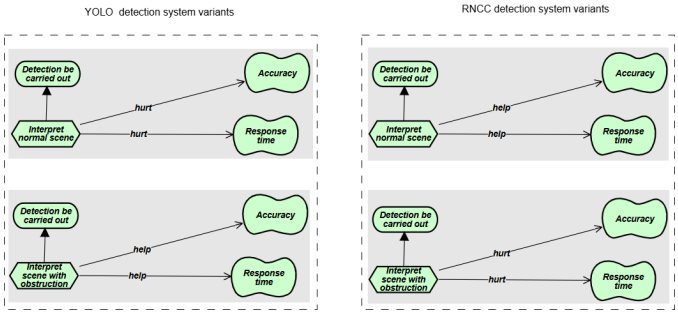

Fig. 4. YOLO and RCNN variants and their impacts on soft goals.

In this way, we can characterize the YOLO detector variants

Paper	Variability	Goal Modeling: Automatic Interpretation	Pragmatic Interoperability
[11] *Goal Model Extraction from User Stories Using LLMs*	Interprets variable granularity in user stories.	GPT-4 extracts actors, goals, and softgoals from user stories.	Indirect — improves shared understanding but lacks runtime support.
[12] *GPT-4 for Goal Model Creation*	Variability in prompts and model outputs.	GPT-4 interactively builds GRL models using different prompt settings.	Indirect — supports modeling quality, not runtime coordination.
[19] *LLMs to Detect Variability in Requirements*	Core — detects variability in behavioral and non-functional requirements.	Does not include goal interpretation or modeling.	Indirect — clarifies requirements but no execution-level alignment.
[10] *Traceability from Security Requirements to Goal Models*	Traceability focus; variability implicit.	LLM (GPT-3.5) links security requirements to GRL goals.	Design-time alignment between intent and specification.
Our Proposal *LLM-Based i* Model Variant Analysis for Pragmatic Interoperability*	Detects variation points and evaluates variants.	Uses LLM to interpret i* SR models, analyze tasks, softgoals, and context.	Runtime-focused — improves operational collaboration via intention interpretation and variant selection.

TABLE I
COMPARISON OF CONTRIBUTIONS ACROSS VARIABILITY, GOAL MODELING AUTOMATIC INTERPRETATION, AND PRAGMATIC INTEROPERABILITY

as follows:

- When the detector interprets a normal scene, it achieves the detection goal, but the interpretation task accuracy and response time are negatively affected.
- When the detector interprets a scene with obstruction, it achieves the detection goal, and the interpretation task accuracy and response time are positively affected.

III. RELATED WORK

Large Language Model (LLM) application to goal-oriented requirements engineering has recently been promoted as a viable way to effectively facilitate automation, especially to reason about user intentions and boost system interoperability. A series of studies have been made on using LLMs to assist in generating, investigating, or matching goal models with requirements documents.

Siddeshwar et al. [11] propose using GPT-4 to distill goal models from user stories in agile settings. Their approach leverages iterative prompt engineering and few-shot learning to generate GRL models, e.g., actors, goals, and soft goals. This work makes a valuable contribution by handling variability in user story writing by stakeholders using granularity alignment and extracting implicit intentions. However, the final products focus on design-time formalization rather than runtime execution or coordination of multiple systems.

Chen et al. [**?**] also evaluate GPT- 4's skills in constructing GRL goal models but emphasize the influence of different schemes of prompting and domain competence. While they confirm that GPT-4 has a sufficient understanding of modeling aspects, their effort is more about checking the syntactical and structural validity of the models generated. It does not concern itself with runtime variability and system interoperation.

Fantechi et al. [19] resort to discovering the variability of textual requirements. They assess the performance of LLMs to extract functional and non-functional variability in natural language documents. Whilst the paper is not directly about goal modeling, its findings are transferable to explaining ambiguous/divergent requirements that would hinder interoperability at the point of integrating systems.

Hassine [10] also contributes to traceability by suggesting that links between security requirements specified can be traced automatically using natural languages and GRL goal models. With a Zero-Shot prompting method and GPT-3.5, the solution has high recall and precision to bind requirements to goals. Aligning at the security requirements and design model levels is primarily its focus. However, the method works mainly at design time and does not extend beyond that to cover decision-making at runtime or context-aware coordination.

Our work advances the state of the art by shifting the focus to pragmatic interoperability using dynamic interpretation of goal models with the assistance of LLMs. We integrate i* Strategic Rationale models with GPT-3.5 to extract independent variation points and reason about their variants with the help of contextual background knowledge. That makes the dynamic at-runtime selection of adequate system behaviors feasible so that the interpreted result concerns the desired effect in the context. Handling at-runtime decision-making, our solution bridges a gap in the literature with one step further from static model generation to operational intention alignment on diversified systems. While earlier work proved that LLMs assist in producing goal models, traceability, and variability detection, our proposal is the first to encompass all three areas—variability management, understanding of the goal model, and runtime pragmatic interoperability—under a single, automated activity.

Table I compares the literature work to our purpose. In summary, we use LLM, which aims to solve pragmatic interoperability in runtime and directly supports operational collaboration.

IV. THE RESEARCH DESIGN

Our objective in defining this research project is to show that the proposed process can detect variation points and

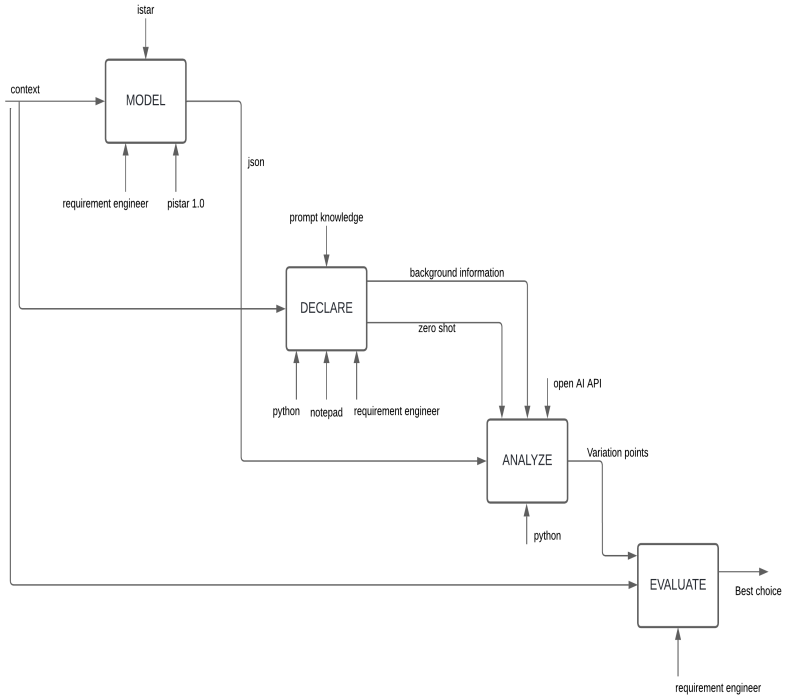

Fig. 5. The method as described by an SADT Actigram.

analyze variants to select the appropriate detector systems. This purpose is validated through automatic detection of the two variation points of the model illustrated in Fig. 3 and automatic finding and analyzing the four variants (Fig. 4) produced by the variation points, to solve the pragmatic problem illustrated in Figure 1 (b).

In Figure 5, the SADT actigram [14] shows the four main activities that reflect our research design:

- MODEL: Given the i*modeling language, a given context is modeled using our version of Pistar [15] to reflect the i* [16] original semantics, and a JSON file is generated to allow further analysis.
- DECLARE: Given our knowledge of prompting LLMs, and the context given, we write different context descriptions (background information) and queries. This activity is supported by Python and a text editor.
- ANALYZE: Each context and query generated in the DECLARE activity and the OpenAI API control the analysis of the JSON file generated in the MODEL activity. A Python script is used for activating the responses provided by the LLM generating variation points.
- EVALUATE: Using recall and precision and a gold standard produced by two of the co-authors the variation points are evaluated through an Experiment (Section 5) to compare the different strategies used in the DECLARE activity.

V. The Experiment

We designed a controlled experiment evaluating LLM performance in detecting independent variation points in SR models.

A. Experiment design

Our goal is to evaluate the LLM's performance in detecting variation points.

The independent variables are a zero-shot prompt (Fig. 7) with contextual prompt variations. Zero-shot prompting does not provide explicit examples or demonstrations of how to complete the task. The model relies entirely on the wording of the prompt to understand the task. However, contextual prompting can provide additional information (not examples) to guide the model's understanding, which can improve zero-shot performance by providing relevant background.

```
# Combine background information with the goal model and with the prompt task.
prompt = (f"Background: {raw_background}\n\nHere is a goal model: {goal_model_str}."
          " Detect independent explicit designed variation points in this goal model.")
```

Fig. 7. The python code prompt.

The dependent variables are the precision and recall metrics to evaluate how the model performs in terms of detecting correct information.

There is one control group. It uses a goal model for image interpretation analysis. This group uses no background

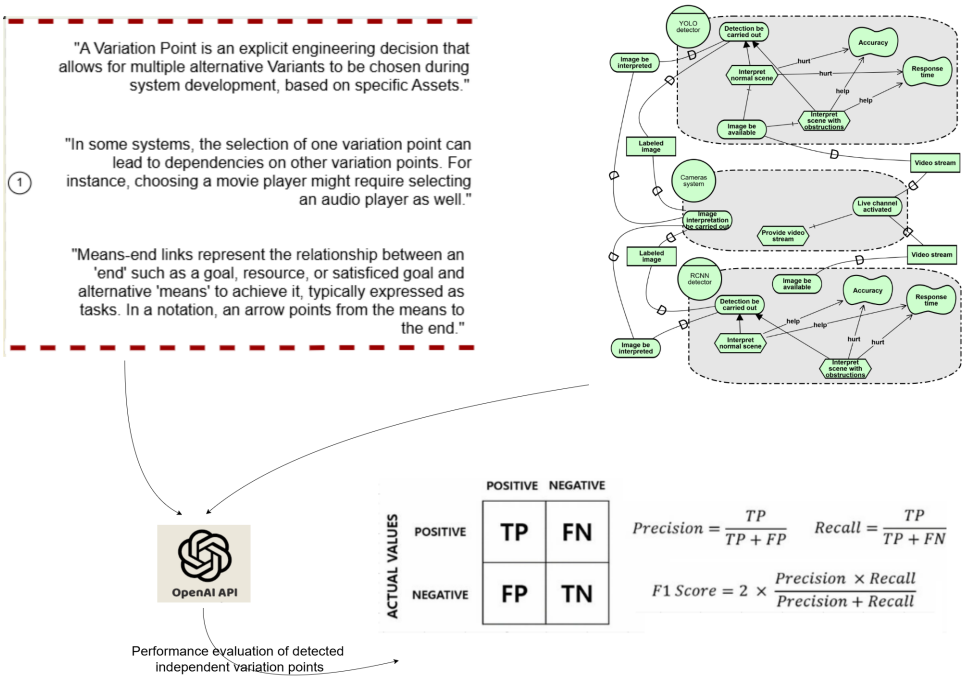

Fig. 6. A process to detect variation points in SR models using ChatGpt.

information. It is a baseline with no variations in prompts or data against which to compare the experimental group.

Each model and their variations represent different experimental groups.There are four variations of background information for each model. The goal is to analyze whether the performance in detecting independent variation points of each prompt in the experimental group is better than the performance of each prompt in the control group. The performance analysis of the control group and that of the experimental group used descriptive analysis with metrics like precision and recall to determine the LLM's performance under different conditions.

Figure 6 illustrates the variation point detection process through the Openai API gpt-3.5-turbo-0125. This model is cheaper than the GPT-4. Openai API is used for prompting. Context (background) information and the SR model are given to the API in order to detect independent variation points (IVP). This context information guides the LLM to better understand the task at hand (detect IVPs). Different sets of background information were used. Precision, recall and f-measure are calculated based on the true positives, true negatives, false positives, and false negatives identified by comparison with the gold standard.

B. The experiment execution

To execute the experiment, we invoked the API, with $maxtokens = 400$ and $temperature = 0.2$. You can find the source code on GitHub.[1]. First, we provided the prompt with no background information (empty Txt file) and the image interpretation goal model (textual model representation in JSON file). This textual model represents the graphical model based on Figure 2. This prompt data retrieval represent the control group. This configuration returned all intentional element described in Table II.

Element Type	Intentional elements
A	Interpretation be carried out
B	Interpret normal scene
C	Interpret scene with obstructions
D	Accuracy
E	Response time
F	Image be available

TABLE II
ELEMENT TYPES FOUNDED IN SR GOAL MODEL.

The IVP in Figure 3 is the Gold triplet = {A, B, C} in YOLO and RCNN agents. This IVP is our gold standard [17]. The API found the exact independent variation point triplet in retrivel 5 (with background 5), as reported in Table III.

In our evaluation case, the system was required to return elements A, B, and C in every retrieval attempt. The strict evaluation policy was employed, according to which the retrieval was regarded as correct (True Positive) only if it contained exactly the elements A, B, C, i.e., no omissions of the elements, no extraneous elements. Any retrieval, which had at least one of the required elements omitted, was regarded as a False Negative, and any retrieval, which contained all the required elements but also had some extraneous, irrelevant elements, was regarded as a False Positive.

Based on data obtained in Table III, LLM obtained Precision $= 0.25$, Recall $= 0.50$ and F1-score $= 0.\overline{3}$.

[1]https://github.com/FigueiredoRoberto/llmistar

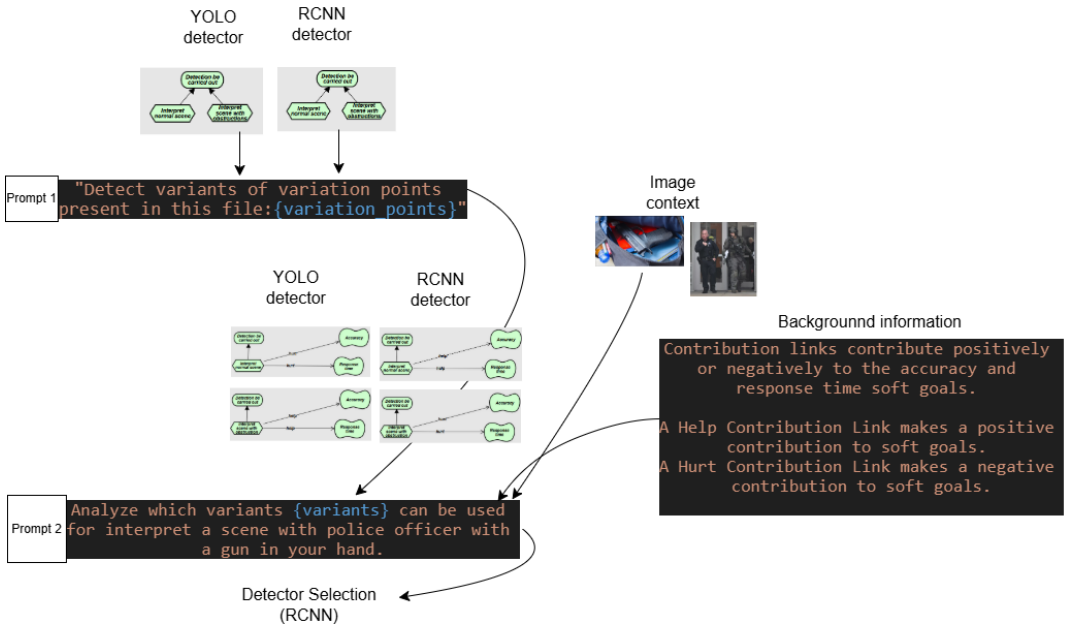

Fig. 8. A process to select detectors based on image context.

Retrieval #	Retrieved Set	Evaluation
1	{A, B, C, D, E, F}	False Positive (extra elements)
2	{B, C}	False Negative (missing A)
3	{A, B, C, D, E, F}	False Positive (extra elements)
4	{A, B, C, D, E}	False Positive (extra elements)
5	{A, B, C}	True Positive (exact match)

TABLE III
RETRIEVED SET FOR EACH BACKGROUND.

VI. SELECTING DETECTORS BASED ON VARIANTS ANALYSIS

Given that proper background prompts do allow for IVP identification, we show results of the detector selection analysis based on the context of each gun image and variant analysis.

As illustrated in Figure 8, the API was provided with: Prompt 1, the independent variation points of each detected actor, and the task of detecting correspondent variants. As such, the API detected the correct variants.

Given the detected variants, background information about contribution links, and image context, we designed one new Prompt 2 to determine which detector variant is best for each image context. The API found the best choices as shown in Table IV.

Context	Detector variant chosen
Scene with a gun inside a backpack	YOLO detector variant
Scene with police officer with a gun	RCNN detector variant

TABLE IV
CONTEXT ANALYSIS AND CHOSEN DETECTORS BY LLM.

VII. DISCUSSION

In this section, we will evaluate the results presented in Tables III and IV, which relate to the detection of IVPs (independent variation points) in the image interpretation goal model and the detectors selected for each image context.

Concerning background variations (Tab. III), we conclude that in background 1, the performance in the control group is ineffective, for which the model fails to identify the exact match of the independent variation points. This was expected, given that the control group lacks the necessary information or context to understand the independent variation points in the SR model. In Background 2, two elements of the triple were retrieved, representing an improvement in the returned data. Although element A does not appear, other irrelevant elements were not returned. This may indicate that the background information in this group helps filter irrelevant intentional elements for the model. For Background 3, the model recalled all the elements of Background 1. That means that while the model is good at retrieving most items related to IVP, there is much over-selection, contributing to several false positives. In Background 4, element F was not returned. This suggests that the background probably provided a better context. In Background 5, the results match precisely with the gold triplet, indicating that contextual data contributed to improving precision and recall. This, in its entirety, shows that Backgrounds 5 present more helpful information about the object detection SR model. The precision of 0.25 indicates that only one in four retrievals that included the triplet A,B,C did so without including extra, irrelevant elements. On the other hand, the recall value of 0.50 shows that the system was able to retrieve the correct triplet in half of the expected cases. The F1-score of 0.33 confirms that while the system captures some relevant information, its tendency to include unnecessary elements significantly lowers the quality of the retrieved results when strict relevance is demanded.

About the chosen detectors by LLM (tab. IV), we con-

clude that the variants between detectors are unique, and it allowed the LLM choices based on contexts. This contributes to supporting pragmatic interoperability and resolving the problem illustrated in Fig. 1 (b). Background information from prompt 2 (Fig. 8), based on concepts of contribution links, was decisive for LLM to select the appropriate detector.

Although our argumentation is based on a small example, it shows that the merging of LLM with intentional modeling provides a path to have automated assistance for supporting pragmatic interoperability. The role of intentional modeling is the possibility of having points of choice, variation points. Identifying these points is key to analyzing modeled possibilities and choosing the proper variants, as performed using the LLM. As identified in previous work on GORE modeling [12], and shown here, the proper requirements (prompts) are essential to achieve better results.

Of course, dealing with large models will impose more load on LLM based IVP identification, and affect the efficiency of this type of solution. However, different alternatives could be used to address the issue, including improving the precision, as well as, turning the intentional model prior to the proposed process.

VIII. Conclusion

The automatic analysis of intentional models using an LLM demonstrates a promising path toward pragmatic interoperability. Our strategy, centering attention on detecting variation points through the use of LLM, allowed us to experiment with the importance of proper background prompts. Our work followed a research design to support automation with a modeling tool and Python scripts to analyze the different strategies. We used recall and precision measures to evaluate these strategies.

The results using LLM to identify independent variation points are positive, however, detection precision needs to be improved, thus allowing for the analysis of large models. Although the samples we have used are not large models, the strategy is scalable once enough resources are provided. The level of understandability by GPT of your goals was sufficient, but more tuning and experiments may improve our strategy.

As for future work, besides tuning, and larger intentional models, we plan to use Retrieval Augmented Generation (RAG) [21] to supplement the general LLM. Expanding the exploration of different scripts will allow for the merging of other sources to improve the performance of the automation of variation point detection and variant selection.

Acknowledgment

Leite acknowledges the partial support from CNPq. This material is partially based upon work supported by the FAPESB under grant TIC 0002/2015. This material is partially based upon work supported by CAPES Financial code 001.

References

[1] Elivaldo Lozer Fracalossi Ribeiro, Erasmo Leite Monteiro, Daniela Barreiro Claro, and Rita Suzana Pitangueira Maciel, "A Conceptual Framework for Pragmatic Interoperability," in *Proceedings of the XV Brazilian Symposium on Information Systems (SBSI '19)*, Aracaju, Brazil, 2019, Art. no. 36, 8 pages, Association for Computing Machinery, New York, NY, USA. DOI: 10.1145/3330204.3330246.

[2] Christian H. Asuncion and Marten J. Van Sinderen, "Pragmatic interoperability: A systematic review of published definitions," in *EAI2N 2010, Held as Part of WCC*, 2010, pp. 164–175.

[3] Bachmann, Felix, et al. "A meta-model for representing variability in product family development." International workshop on software product-family engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003.

[4] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems 28 (2015).

[5] Redmon, J., "You only look once: Unified, real-time object detection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[6] Krishna Ronanki, Christian Berger, and Jennifer Horkoff, "Investigating ChatGPT's Potential to Assist in Requirements Elicitation Processes," in *2023 49th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2023, pp. 354–361. IEEE. DOI: 10.1109/SEAA60479.2023.00061.

[7] Archana Tikayat Ray, Bjorn F. Cole, Olivia J. Pinon Fischer, Anirudh Prabhakara Bhat, Ryan T. White, and Dimitri N. Mavris, "Agile Methodology for the Standardization of Engineering Requirements Using Large Language Models," *Systems*, vol. 11, no. 7, p. 352, 2023. MDPI. DOI: 10.3390/systems11070352.

[8] Gärtner, Alexander Elenga, and Dietmar Göhlich. "Automated Requirement Contradiction Detection through Formal Logic and LLMs," in *Automated Software Engineering 31.2 (2024): 49.*.

[9] Günter Halmans and Klaus Pohl, "Communicating the variability of a software-product family to customers," *Software and Systems Modeling*, vol. 2, no. 1, pp. 15–36, 2003. Springer-Verlag. DOI: 10.1007/s10270-003-0019-9.

[10] Jameleddine Hassine, "An LLM-based Approach to Recover Traceability Links between Security Requirements and Goal Models," in *28th International Conference on Evaluation and Assessment in Software Engineering (EASE 2024)*, Salerno, Italy, 2024. ACM. DOI: 10.1145/3661167.3661261.

[11] Vaishali Siddeshwar, Sanaa Alwidian, and Masoud Makrehchi, "Goal Model Extraction from User Stories Using Large Language Models," in *Proceedings of the 17th International Conference on the Quality of Information and Communications Technology (QUATIC)*, 2024, pp. 269–276. Springer Nature Switzerland AG. DOI: 10.1007/978-3-031-70245-$7_1$9.

[12] B. Chen, K. Chen, S. Hassani, Y. Yang, D. Amyot, L. Lessard, G. Mussbacher, M. Sabetzadeh, and D. Varró, "On the Use of GPT-4 for Creating Goal Models: An Exploratory Study," in *Proc. of the IEEE 31st International Requirements Engineering Conference Workshops (REW)*, IEEE, pp. 256–263, 2023, doi: 10.1109/REW57809.2023.00052.

[13] Stan Bühne, Günter Halmans, and Klaus Pohl, "Modeling dependencies between variation points in use case diagrams," in *Proceedings of the REFSQ 2003*, 2003. University of Duisburg-Essen.

[14] Douglas T. Ross, "Structured analysis (SA): A language for communicating ideas," *IEEE Transactions on Software Engineering*, vol. 1, pp. 16–34, 1977. IEEE.

[15] João Pimentel and Jaelson Castro, "Pistar tool–a pluggable online tool for goal modeling," in *2018 IEEE 26th International Requirements Engineering Conference (RE)*, 2018, pp. 498–499. IEEE.

[16] Eric S.K. Yu, "Modelling Strategic Relationships for Process Reengineering," Ph.D. dissertation, University of Toronto, 1995. Tech. Report DKBS-TR94-6, Dept. of Computer Science, University of Toronto.

[17] Daniel M. Berry, "Requirements Engineering for Artificial Intelligence: What is a Requirements Specification for an Artificial Intelligence?" *Proceedings of the 2022 Requirements Engineering for AI (REFSQ)*, 2022. IEEE. Cheriton School of Computer Science, University of Waterloo.

[18] Daniel Gross, Arnon Sturm, and Eric Yu, "Towards Know-how Mapping Using Goal Modeling," *Proceedings of the 6th International i* Workshop (iStar 2013)*, vol. 978, pp. 115–120, 2013.

[19] Alessandro Fantechi, Stefania Gnesi, Laura Semini, and Jill Tamanini, "Exploring LLMs' ability to detect variability in requirements," in *Proceedings of the REFSQ 2024 Research Track*, Essen, Germany, 2024, pp. 178–188. Springer. DOI: 10.1007/978-3-031-57327-9$_1$1.

[20] Figueiredo, R. de C., Claro, D. B., Maciel, R. S. P., Leite, J. C. S. do P.: Using i* For An Early Analysis Of Interoperability Requirements. In: Proceedings of the Workshop on Requirements Engineering (WER 2024), pp. 1–14. (2024)

[21] Wenqi Fan *et al.*, "A survey on RAG meeting LLMs: Towards retrieval-augmented large language models," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024.

[22] Daniel M. Berry, "Evaluation of tools for hairy requirements engineering and software engineering tasks," Tech. Report, School of Computer Science, University of Waterloo, 2017.