Bridging Natural Language and Formal Specification — Automated Translation of Software Requirements to LTL via Hierarchical Semantics Decomposition Using LLMs

ASE 2025
Sun 16 - Thu 20
Seoul, South Korea

Zhi Ma[1], Cheng Wen[1], Zhexin Su[1], Xiao Liang[1], Cong Tian[1]* , Shengchao Qin[1] and Mengfei Yang[2]
[1]Xidian University, [2]China Academy of Space Technology

Motivation

- Automating NL-to-LTL translation is vital for scaling formal verification in safety-critical domains.
- Existing rule-based, learning-based, and direct LLM approaches fail to handle industrial-scale complexity and ambiguity.

Zero-Shot Prompt: You are an expert in the field of formal verification, proficient in the rules and usage of LTL formulas. Please translate the following natural language requirement into LTL formulas.

Req01: In the inertial navigation valid mode, the output will be set to reverse navigation computation, or the output will be set to low velocity low altitude navigation computation, or the output will be set to high velocity navigation computation, or the output will be set to high altitude navigation computation.	**Req02**: Globally, if the navigation system computes angles using dual inertial navigation modules, the input attitude angle shall be the weighted average of angles from module 1 and module 2 (weights 0.6 and 0.4), unless the system switches to single module or GPS signal is restored.
LTL01 by GPT-4o: G (valid_mode → (output = reverse_nav ∨ output = lowVlowH_nav ∨ output = highV_nav ∨ output = highH_nav))	**LTL02 by GPT-4o:** G ((Dual_INU_Active ∧ ¬Single_Module ∧ ¬GPS_Restored) → (Input_Attitude_Angle = 0.6 * Angle_Module1 + 0.4 * Angle_Module2))
Ground Truth: G (valid_mode → (F)output = reverse_nav ∨ (F)output = lowVlowH_nav ∨ (F)output = highV_nav ∨ (F)output = highH_nav))	Ground Truth: G (Dual_INU_Active (⟶) (Input_Attitude_Angle = 0.6 * Angle-Module1+ 0.4 * Angle_Module2) (U) (Single_Module (∨) GPS_Restored))

Fig. 1. **Examples showcasing the discrepancy between LLM generated LTL and ground truth for complex requirements.** The red-circle indicates errors in the LLM's output due to missing details, contrasted with the manually corrected specification.

- **Challenge 1: Missing Implicit Semantics.**
GPT-4o misses the 'Eventually' (F) operator in "will be set".

- **Challenge 2: Flattened Logic.**
GPT-4o misinterprets "unless" and fails to generate the 'Until' (U) operator.

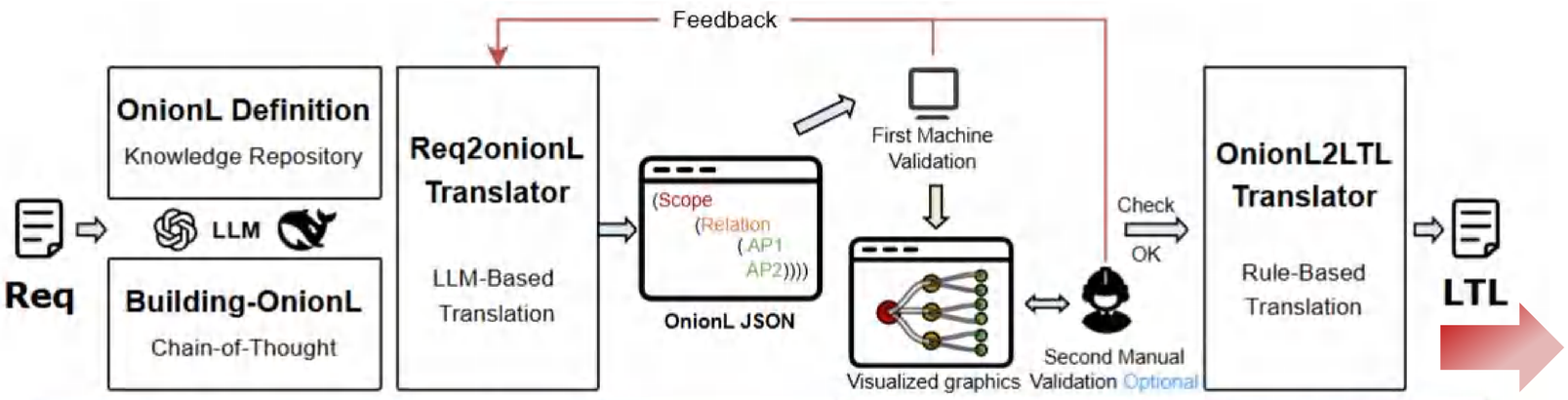
Email:
{mazhi, wencheng}
@xidian.edu.cn

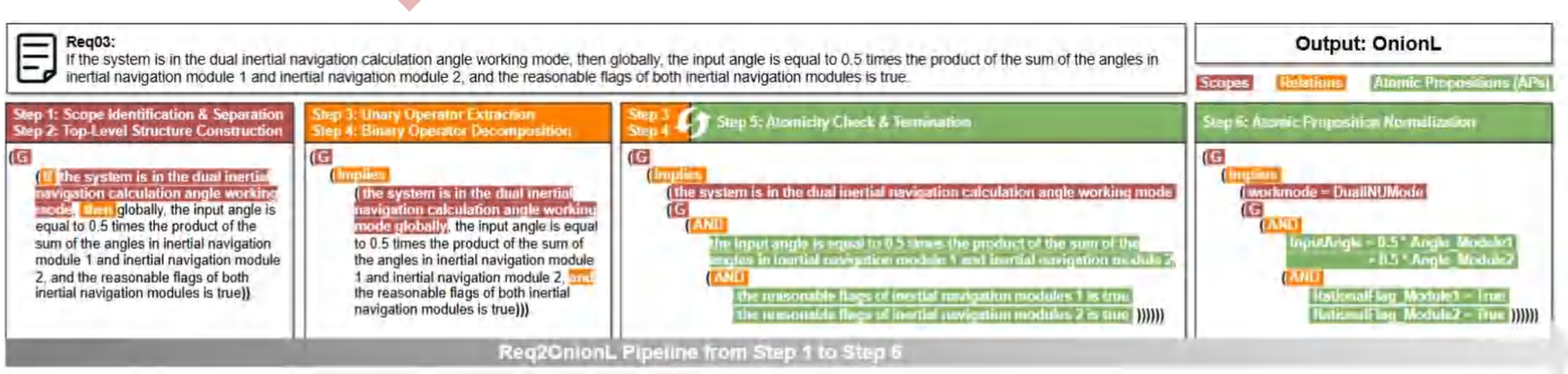
Contribution

- **OnionL:** A novel hierarchical intermediate representation (IR) bridging NL and LTL.
- **Hierarchical Semantic Decomposition:** A two-stage, prompt-driven algorithm (BUILDING-ONIONL).
- **REQ2LTL Framework:** Combines LLM decomposition with rule-based translation. Achieves 88.4% accuracy.

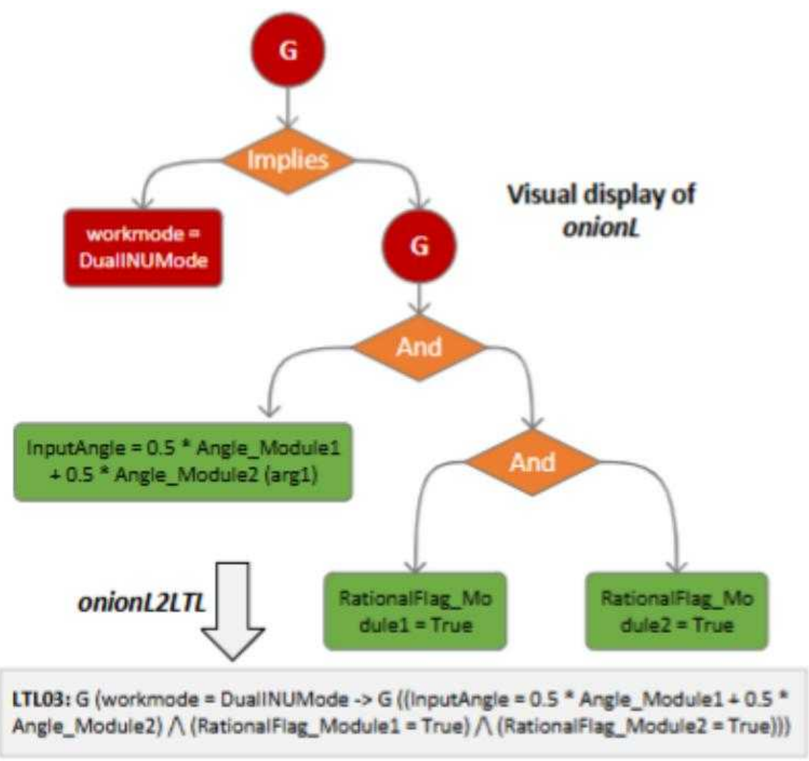
Req2LTL Framework

REQ2LTL is a two-stage pipeline:





Stage 1 (Req-to-OnionL): An LLM, guided by a knowledge repo and Chain-of-Thought (CoT), translates NL to a structured OnionL JSON.



Stage 2 (OnionL-to-LTL): The OnionL JSON is validated (machine + optional manual) and then deterministically translated into LTL.

Evaluation

RQ1: How does REQ2LTL perform on standard academic benchmarks?
√ **Outperforms others on academic benchmarks**

TABLE II
PERFORMANCE COMPARISON ON ACADEMIC BENCHMARKS (LIFTED)

Method	Domain	Binary Acc. (%)	BLEU Score
Req2LTL	Circuit	95.30	**0.97**
	Navigation	94.50	**0.97**
	Office Email	**96.70**	**0.98**
NL2TL	Circuit	**96.20**	0.96
	Navigation	**96.50**	**0.97**
	Office Email	96.00	0.96
NL2LTL	Circuit	90.50	0.93
	Navigation	89.90	0.93
	Office Email	90.80	0.93
NL2SPEC	Circuit	90.80	0.93
	Navigation	89.70	0.93
	Office Email	91.20	0.93

RQ2: How accurately and reliably can REQ2LTL translate real-world industrial requirements?
√ **Highest accuracy on industrial requirements**

RQ3: What is the contribution of each component in REQ2LTL?
√ **All components contribute significantly**

TABLE III
COMPARISON OF PROMPTING STRATEGIES AND ABLATION STUDY ON REQ2LTL FRAMEWORK. BOLD AND UNDERLINED VALUES INDICATE THE BEST AND SECOND-BEST RESULTS, RESPECTIVELY.

Setting	Exact Match (%)	Syntax Validity (%)	AP Recall (%)	BLEU Score
Prompt Strategy Comparison				
GPT-4o + Zero-Shot	43.8	89.3	98.5	0.76
GPT-4o + NL2LTL	55.4	91.5	98.4	0.77
GPT-4o + NL2SPEC	56.3	91.9	98.5	0.78
GPT-4o + NL2TL	65.2	94.6	98.1	0.85
GPT-4o + Req2LTL	**88.4**	**100.0**	**99.5**	**0.96**
DeepSeek-V3 + Zero-Shot	34.8	87.5	97.5	0.73
DeepSeek-V3 + NL2LTL	54.0	91.2	98.6	0.78
DeepSeek-V3 + NL2SPEC	54.5	91.9	98.7	0.79
DeepSeek-V3 + NL2TL	58.0	92.9	99.1	0.82
DeepSeek-V3 + Req2LTL	86.6	**100.0**	99.2	**0.96**
Ablation Study on Req2LTL				
D Full Version	**88.4**	**100.0**	**99.5**	**0.96**
A w/o Structured OnionL	65.2	98.2	98.9	0.86
B w/o Stage-wise Decomposition	58.9	94.6	98.1	0.85
C w/o Verification Feedback	84.8	90.2	**99.5**	0.91

RQ4: How effectively does REQ2LTL address structural complexities in industrial requirements?
√ **Reduces errors, ambiguity remains challenging**

TABLE IV
STRUCTURAL COMPLEXITY COMPARISON: INDUSTRIAL VS. ACADEMIC

Metric	Industrial	Academic
Avg. sentence length (tokens)	43.7	15.3
With ≥2-layer nesting	63.2%	9.8%
Avg. number of APs	3.7	2.1

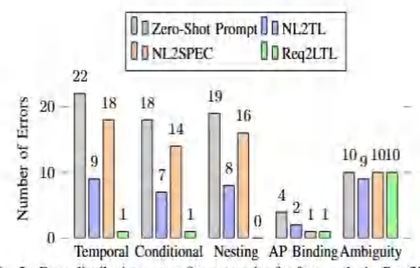


Fig. 5. Error distribution across five categories for four methods. REQ2LTL significantly reduces structural errors, with remaining issues primarily due to ambiguity or contextual omissions.