August 17th, 2026, Montréal, Canada

16th Model-Driven Requirements Engineering Workshop (MoDRE 2026)

Towards Satisfaction Analysis of GRL Model Images Using Vision-Enabled LLMs

Ahmed Hassine
Carleton University
Ottawa, Canada

Carleton University

Jameleddine Hassine
Université du Québec à Montréal (UQAM)
Montréal, Canada

UQÀM

TABLE OF CONTENTS

Goal Modeling

- A goal model is a **conceptual model** that visually represents **goals, their decomposition, and dependencies**.
- Captures stakeholders' **interests, intentions, business goals, and strategies**.
- Models **positive and negative influences** among goals and tasks.
- Supports analysis of **alternatives, trade-offs, conflicts, and dependencies**.
- Most popular goal modeling languages: i*, NFR, TROPOS, **GRL (Goal-oriented Requirements Language, part of the ITU-T User Requirements Notation standard)**

GRL Satisfaction Analysis

- Evaluates **how well stakeholders' goals are satisfied** under a given strategy or scenario.
- A GRL Strategy assigns satisfaction levels to **intentional elements, e.g.,** goals, tasks.
- **Propagates satisfaction values** through decomposition and contribution links.
- Common satisfaction levels range from **Sufficiently Denied (−100)** to **Sufficiently Satisfied (+100).**
- Importance values ([0,100]) weight each element's priority within its actor, used for actor-level aggregation.
- Helps compare **alternative strategies** and identify **trade-offs and conflicts**.

Motivation & Problem

- GRL satisfaction analysis in practice relies on dedicated tooling, e.g., **jUCMNav**, a free Eclipse-based URN environment
- Yet GRL models are most commonly **communicated** and **shared** simply as **graphical diagrams**
  - Analysts often only have the exported image — not the underlying model or the tool environment
- A **tool round-trip** is required just to answer a simple **"what-if" question**: manual re-entry, re-engagement with jUCMNav, and specialized modeling expertise.

Motivation:

> **"Ideally, satisfaction analysis should be possible from nothing more than the diagram image itself."**

Opportunity: Vision-enabled LLMs as a new goal-model analysis tool

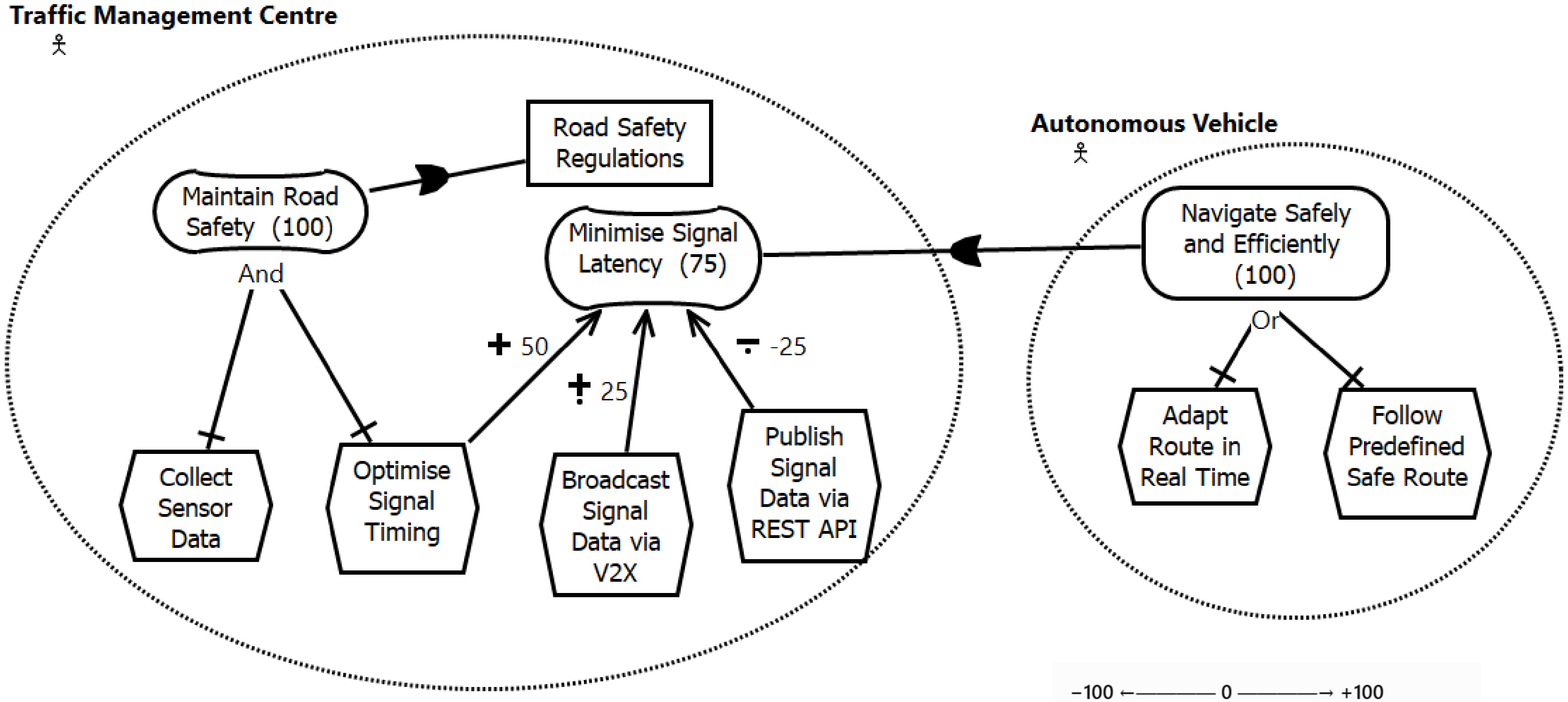
- **Multimodal reasoning**
  - Vision-enabled LLMs jointly process textual and visual inputs
- **Direct analysis from diagrams**
  - If an LLM can faithfully parse a GRL image, no need for the user to transform manually the diagram into another intermediate representation
- **In-context domain knowledge**
  - A GRL reference document can be supplied alongside the image to guide reasoning

Key idea:
LLM receives a GRL diagram image + a GRL notation reference document, enabling satisfaction analysis without manual model transformation or access to the underlying model.

Contributions

- First investigation of a **vision-enabled LLM** performing **GRL satisfaction analysis** directly from **GRL diagram images**
  - From **construct identification** to **satisfaction propagation** and **strategy generation**.
- A publicly available **browser-based prototype tool** with a full replication package, making GRL analysis accessible without formal modeling expertise or dedicated tooling.

Smart City Traffic Management GRL model *



Actor – Holder of intentions: a stakeholder or a system

Softgoal – A quality concern with no clear-cut satisfaction criterion

Goal – An intentional element with a clear satisfaction criterion

Task – An activity or operationalization used to achieve goals and softgoals

Resource – A physical or informational entity required by an intentional element

Links

Decomposition – Refines an element via AND, OR, or XOR relationships into sub-elements

Dependency – One actor's element depends on another actor's element being satisfied

Contribution – Positive or negative impact of one intentional element on another

Quantitative contributions

−100 ← 0 → +100
Break Hurt Help Make
Negative impact Positive impact

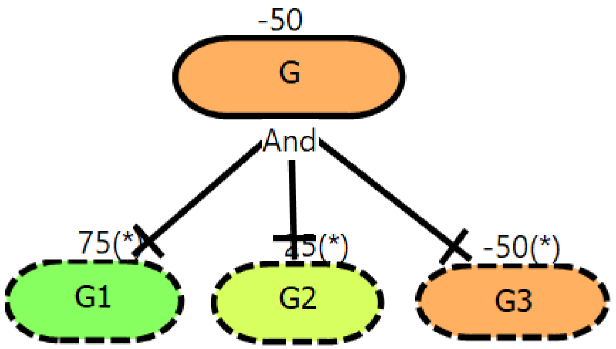
Qualitative

Break | Some Negative | Hurt | Help | Some Positive | Make

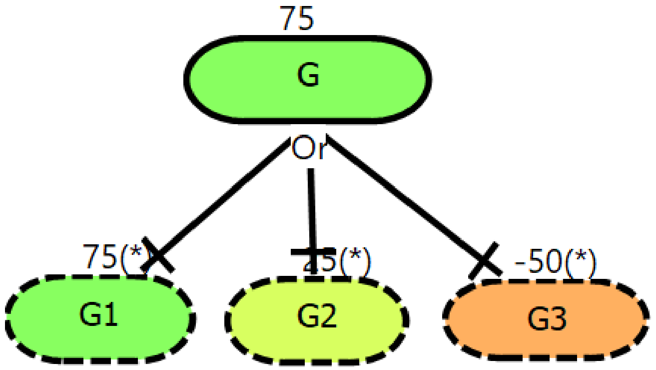
(X) represents the importance of the element: an integer between 0 and 100

* Built using jUCMNav tool

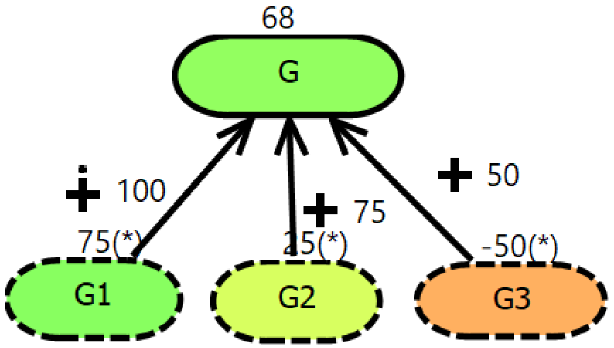
GRL Satisfaction Analysis Computation Examples



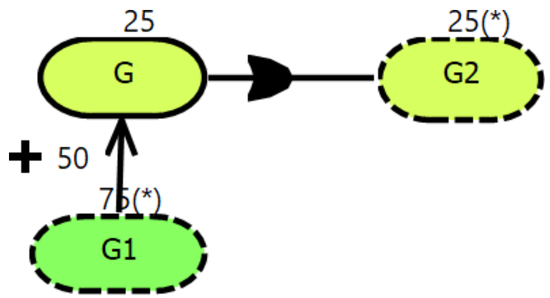
sat(G) = min(G1, G2, G3)



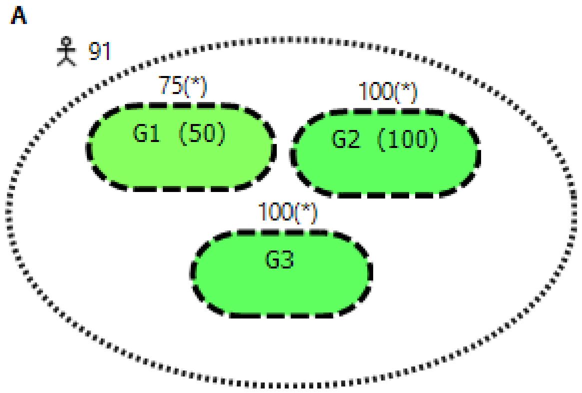
sat(G) = max (G1, G2, G3)



sat(G) = Σ (sat(Gi) × weight_i)
capped at +100 / −100



sat(G) = min(sat(G1) × weight, sat(target))



sat(Actor) = Σ(sat_i × imp_i) / Σ imp_i

sat(A)= (75*50 + 100*100 + 100*0)/(100+50+0) = 91.67

RQ1

Element identification

Can GPT-5.5 accurately identify GRL elements from an image of a GRL model?

RQ2

Satisfaction computation

To what extent can GPT-5.5 accurately compute GRL satisfaction values from an initial strategy?

RQ3

Satisfaction maximization

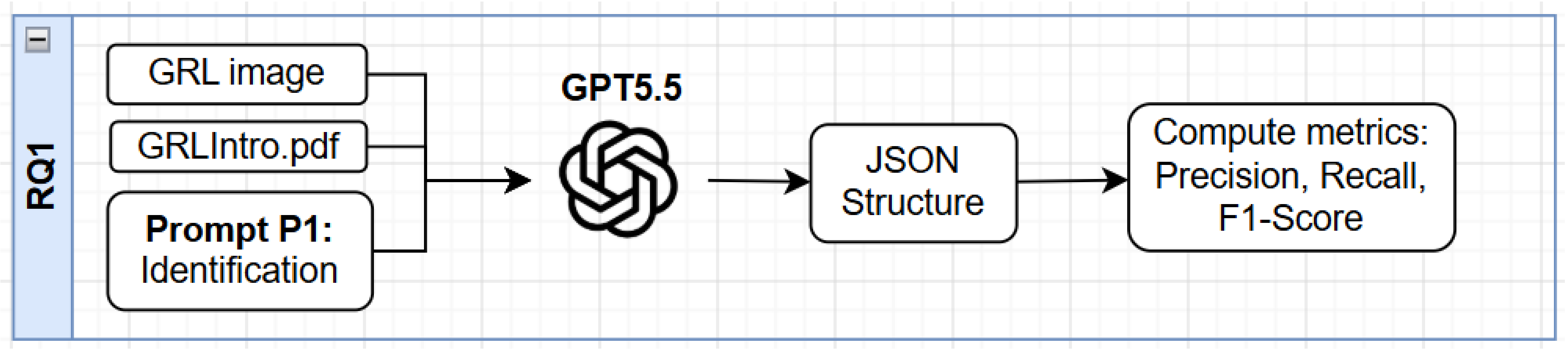
Can GPT-5.5 identify the strategy that maximizes satisfaction of a target element or actor?

RQ4

Threshold satisfaction

Can GPT-5.5 identify a strategy reaching a given threshold, or correctly report it unattainable?

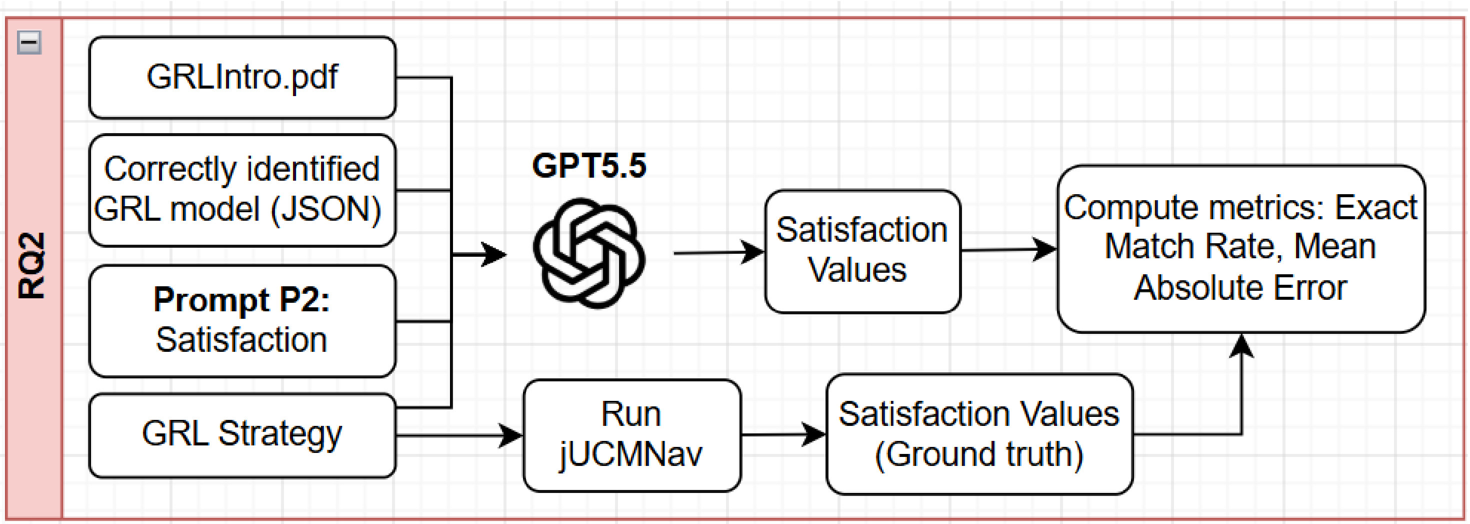
Elements Identification



PROMPT P1

You are a GRL diagram parser. Extract every actor, intentional element, and link from this GRL diagram image. Refer to the attached GRL reference document for notation, element types, and link types. Output a single raw JSON object -- no markdown, no prose, nothing else.
Exact schema:
{"actors":[{"id":"a1","name":"Actor Name"}], "elements":[{"id":"e1","name":"Element Name", "type": "Goal", "actorId":"a1", "importance":100, "decompositionType":"and"}],
"links": [{"id":"l1","type": "Decomposition", "subtype":"and","sourceId":"e2","targetId":"e1"}]}
…

Satisfaction computation



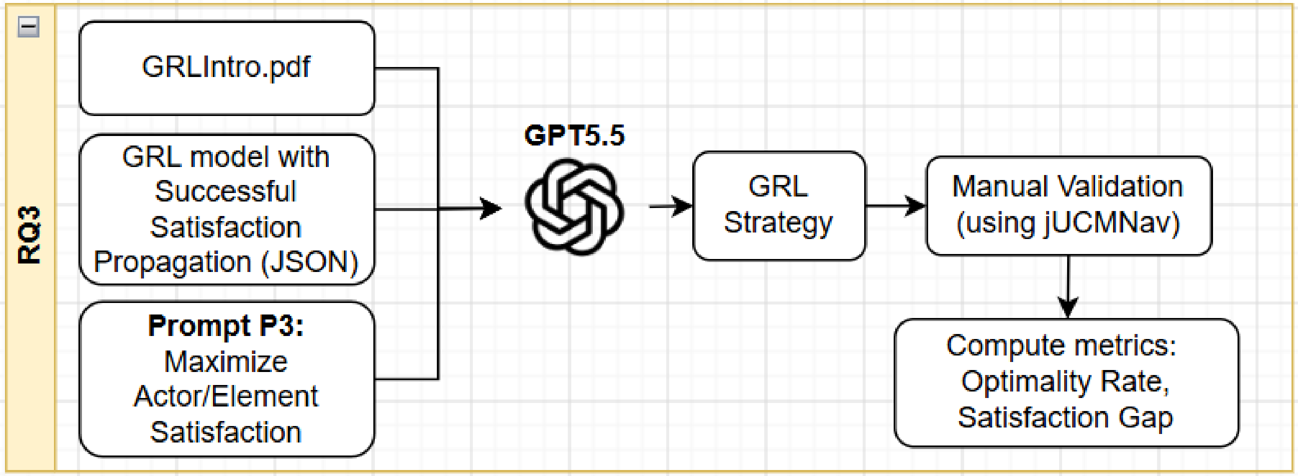
Exact Match Rate (EMR)
Proportion of elements/actors with the correct satisfaction value

Mean Absolute Error (MAE)
Average deviation vs. jUCMNav ground truth across all elements

PROMPT P2

Perform GRL satisfaction value propagation. Apply the propagation rules defined in the attached GRL reference document.
Satisfaction values are integers in [{min};{max}].
INPUT FORMAT:
- [SEED=x]: this element has an initial satisfaction value of x (treat as a fixed leaf input).
- [UNFILLED]: this element has no assigned value; compute it from the graph or omit if unreachable.
…
Respond with ONLY this JSON -- no prose, no markdown fences:
{"propagation":{"ELEM_ID":NUMBER,...}, "overrides":[{"id":"ELEM_ID", "userValue":X, "computedValue":Y, "note":"reason"}], "reasoning":"brief (max 3 sentences)"}

Satisfaction maximization



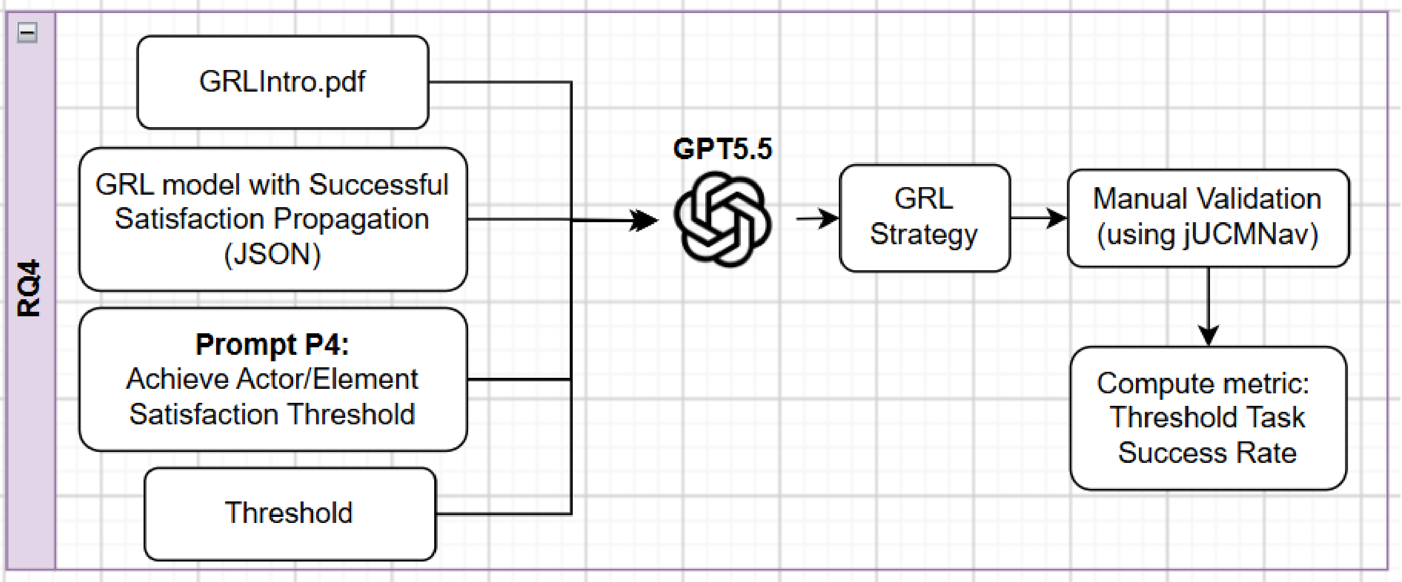
Optimality Rate (OR)
Proportion of cases matching the ground-truth optimal satisfaction value

Satisfaction Gap (SG)
Δs: the shortfall vs. the optimal strategy

PROMPT P3

Given the GRL model below, produce a satisfaction strategy to maximize {scopeDesc}. Apply GRL satisfaction propagation rules from the attached reference document. GRL Model (JSON): {grlJson}
CONSTRAINT: Seed only leaf elements (elements with no incoming decomposition or contribution links), regardless of type.
All non-leaf values must be propagated upward through the GRL links.
Respond with ONLY a raw JSON object -- no markdown, no explanation:
{"seeds":[{"name":"...", "type":"Goal|SoftGoal|Task|Resource|Belief", "actor":"...", "satisfaction":75}], "model":[{"name":"...", "type":"Goal|SoftGoal|Task|Resource|Belief", "actor":"...", "satisfaction":50}]}
Rules:
- seeds: leaf elements only
- In model, group by actor, then order by satisfaction descending within each actor
- Output ONLY the JSON object, nothing else
...

Threshold satisfaction

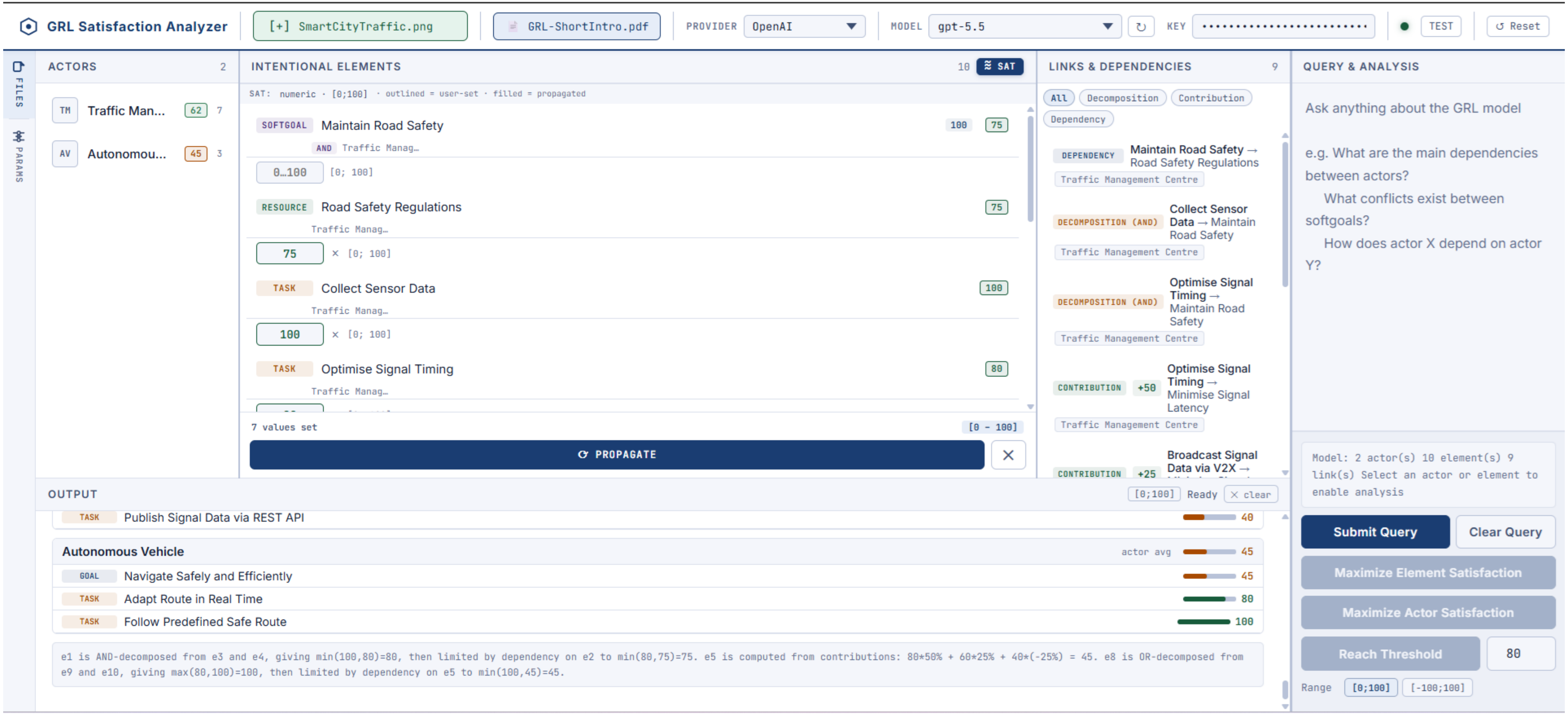


Threshold Task Success Rate

% of tasks where GPT-5.5 correctly reaches the threshold OR correctly reports it unattainable

PROMPT P4

Given the GRL model below, make {scopeDesc} reach or surpass {threshold}. If unattainable, maximize instead and set "threshold_reached": false. Apply GRL satisfaction propagation rules from the attached reference document. GRL Model (JSON): {grlJson}
CONSTRAINT: Seed only leaf elements (elements with no incoming decomposition or contribution links), regardless of type. All non-leaf values must be propagated upward through the GRL links.
Respond with ONLY a raw JSON object -- no markdown, no explanation:
{"threshold_reached":true, "seeds":[{"name":"...", "type":"Goal|SoftGoal|Task|Resource|Belief", "actor":"...", "satisfaction":75}], "model":[{"name":"...", "type":"Goal|SoftGoal|Task|Resource|Belief", "actor":"...", "satisfaction":50}]}
Rules:
- threshold_reached: true if {scopeDesc} reaches ≥{threshold}; false if unattainable
- When false, also include "best_achievable": INTEGER
...



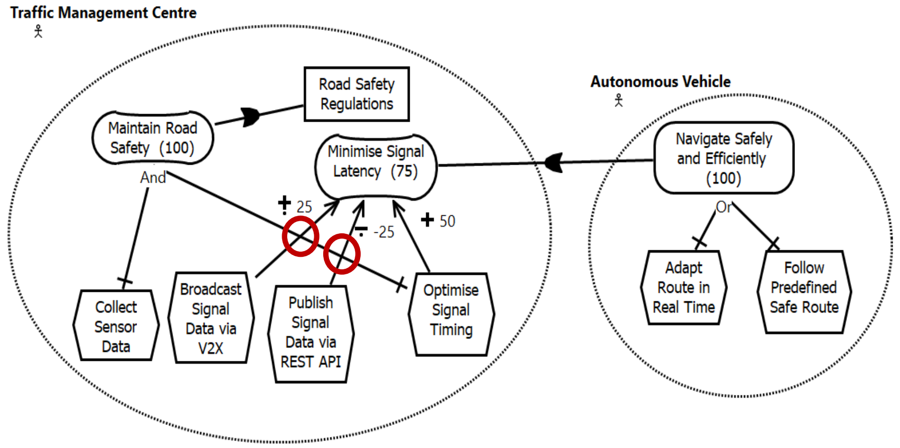
Publicly available at https://github.com/ahmedhassine05/GRL-Satisfaction-Analyzer

Three GRL Models with varying complexity

#	GRL Model	Actors	Elements Total# (G,S,T,R)	Contrib. links (+,–)	Decomp. links (AND,OR)	Dependencies
1	Smart City Traffic System	2	10 (1,2,6,1)	(2,1)	(2,2)	2
2	Electric Vehicle	3	37 (7,11,19,0)	(17,3)	(12,2)	3
3	Online Retail	1	7 (2,2,3,0)	(3,3)	(2,0)	0

Seven image variants (generated using jUCMNav):

- **PNG**: Lossless; preserves sharp edges & text
- **JPG**: Lossy compression; may blur fine details
- **GIF**: Lossless but limited to 256-color palette
- **Dense layout**: Elements and links tightly packed
- **1 intersection:** One pair of crossing links
- **2 intersections:** Two pairs of crossing links
- **Small PNG (50%)**: Reduced-scale version of the original



RQ1 — GRL construct identification (F1-score)

Variant	Model #1	Model #2	Model #3
PNG	1.00	0.99	1.00
GIF	1.00	0.95	1.00
JPG	0.90	0.96	1.00
Dense	0.90	0.97	1.00
Small PNG (50%)	0.95	**0.80**	1.00
1 Intersection	0.95	0.96	1.00
2 Intersections	0.90	0.96	1.00

- Model #1 errors: only goal/softgoal confusions
- Model #2: larger, more structural errors (missing/spurious links)
- Model #2 PNG image has one goal/softgoal confusion => retained for RQ2 to RQ4
- Model #3: perfect (F1=1.00) across all 7 variants
- 50%-scale PNG hurt accuracy most (more than dense layout or intersections)

RQ2–RQ4 — propagation & strategy generation

RQ2: Satisfaction propagation

Three strategies for each model => 9 strategies

Exact Match Rate (EMR): 100 %
Mean Absolute Error (MAE) : 0

RQ3: Maximizing strategies

Nine maximization tasks

Optimality Rate (OR) : 100%
Satisfaction Gap (SG) : 0

RQ4: Threshold strategies

Nine threshold tasks

Threshold Task Success Rate (TTSR): 100%

Once a structurally correct GRL model is recovered from the image, GPT-5.5 performs quantitative propagation and generates optimal threshold-satisfying strategies with perfect accuracy across all evaluated cases.

Threats to Validity

Internal

- Nondeterministic LLM outputs; Mitigated via 5 repeated runs, averaged
- Manually-derived ground truth for RQ3/RQ4, validated by both authors

Construct

- Precision/Recall/F1 don't capture error severity (e.g., goal vs. softgoal confusion)
- OR, SG, and TTSR are quantitative but don't capture strategy quality/diversity

External

- Only 3 GRL models, exported from jUCMNav
- Study limited to GPT-5.5. Results may not generalize to other VLMs

Conclusions

- GPT-5.5 generally identifies GRL constructs from images, but accuracy drops with model complexity & lower resolution.
- Once a structurally correct representation is recovered, satisfaction propagation & strategy generation are perfectly accurate.
- GRL Satisfaction Analyzer makes this workflow available without formal modeling expertise or dedicated tooling.

Future Work

- Larger and more diverse set of GRL models
- Extend to other goal-modeling languages, including i*
- Evaluate additional vision-enabled LLMs for generalizability

THANKS!