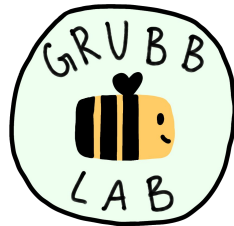
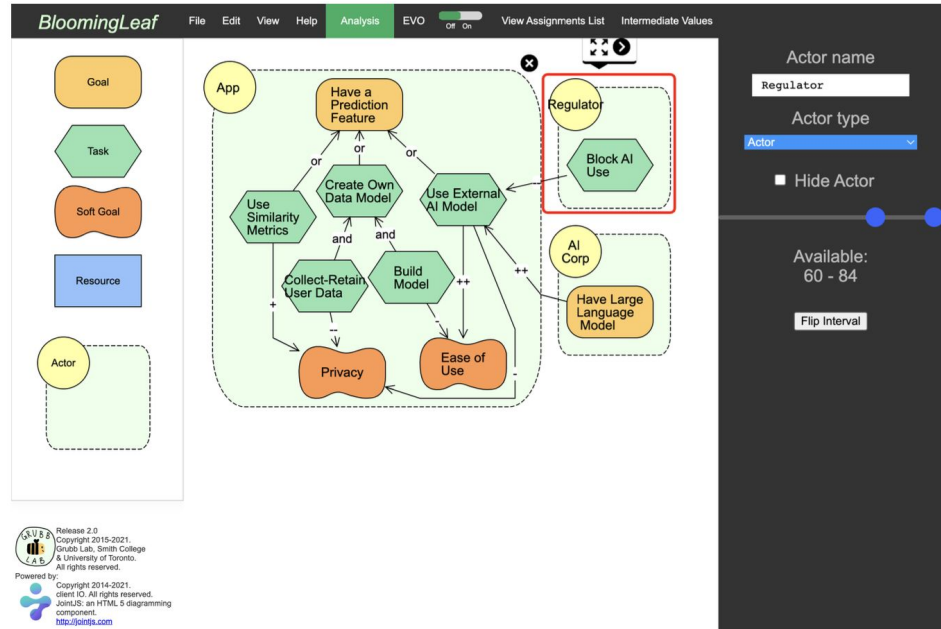


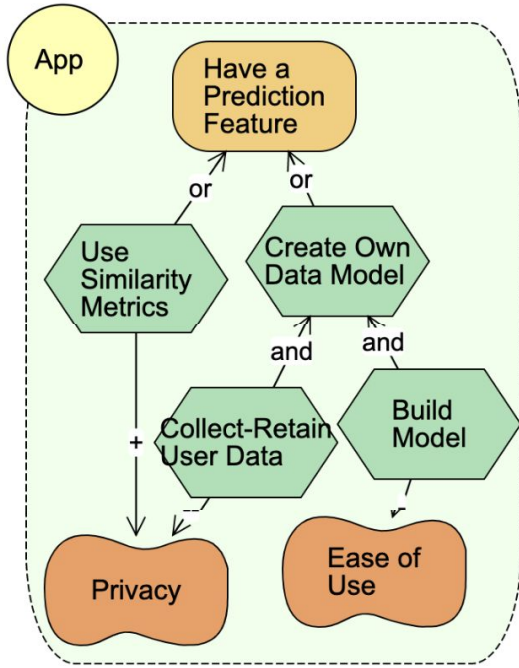
Incorporating Presence Conditions into Goal Models that Evolve Over Time

Xinran Bi and Alicia M. Grubb
Smith College, United States

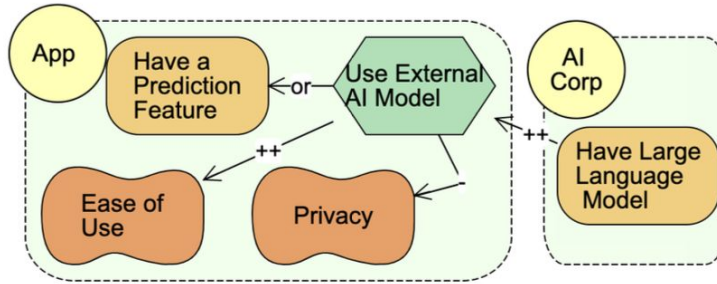


September 5, 2023
Model-Driven Requirements Engineering Workshop (MoDRE)

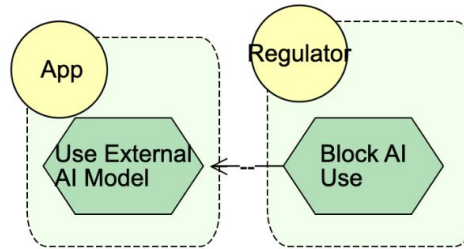
Why presence conditions? Example of the Predictive App



Year 2020

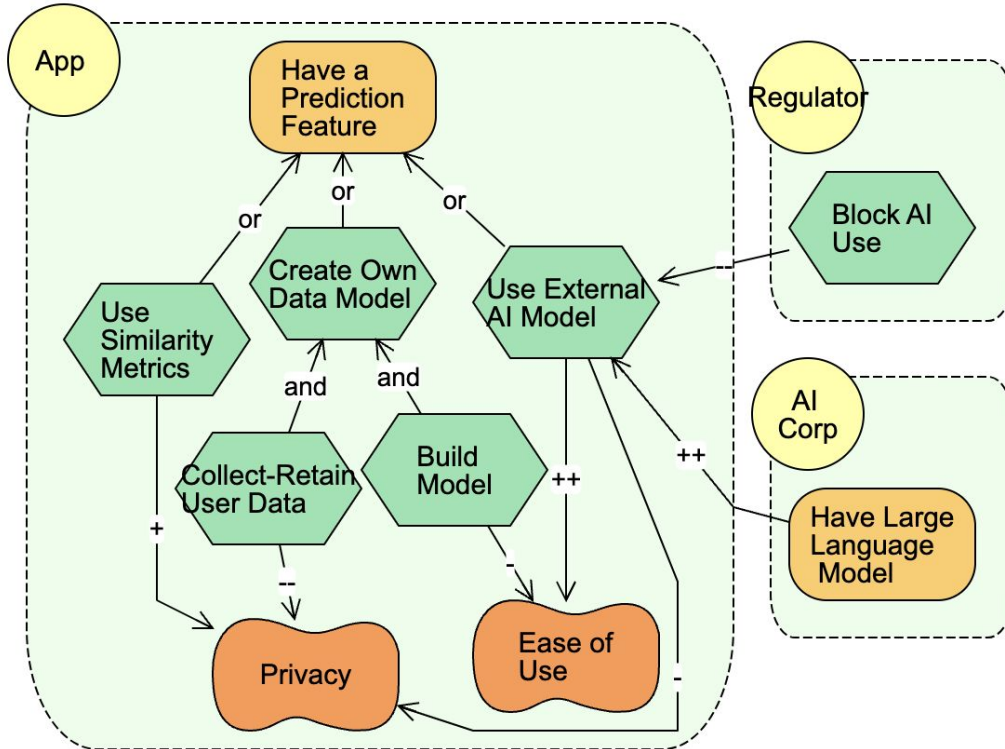


Year 2023

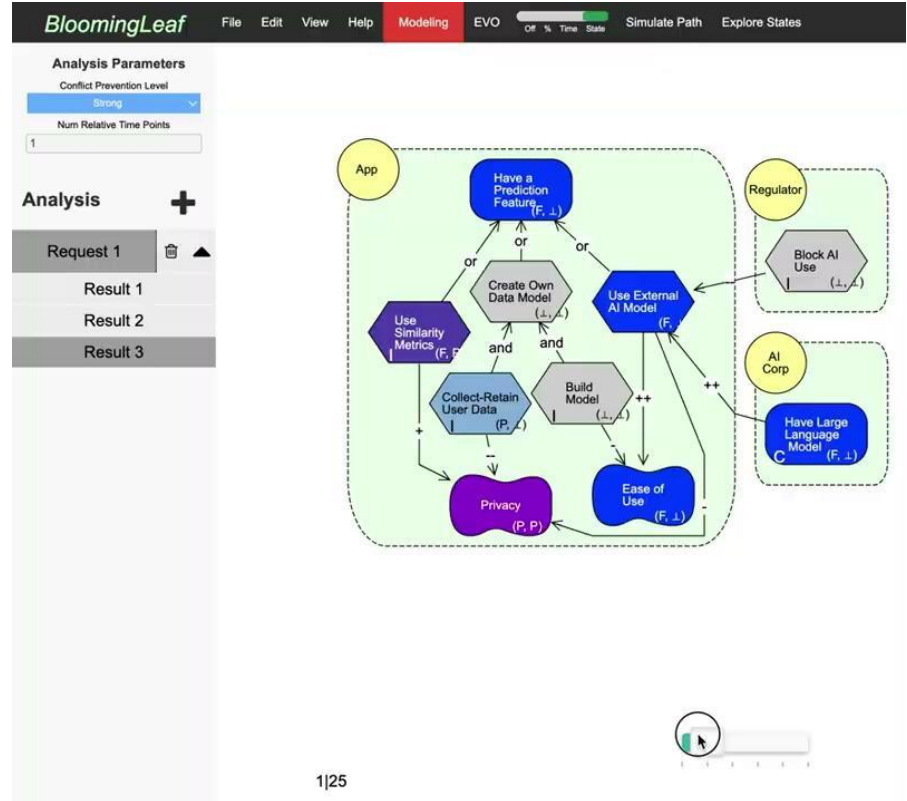
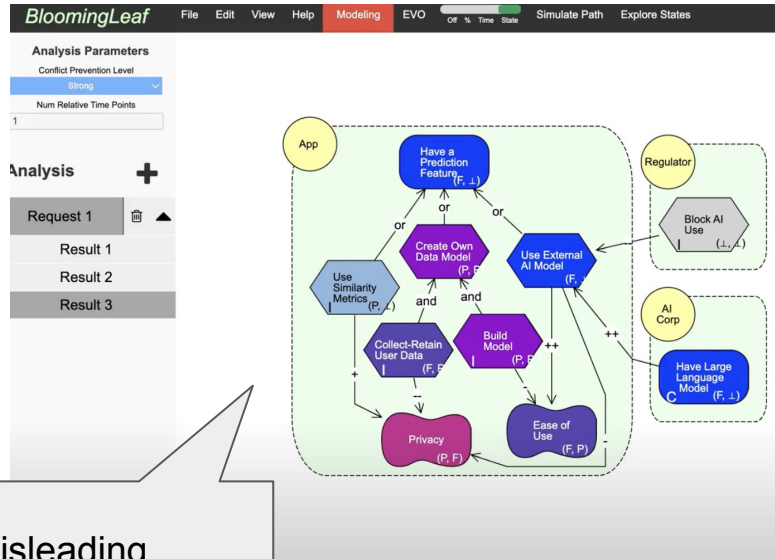


Year 2025

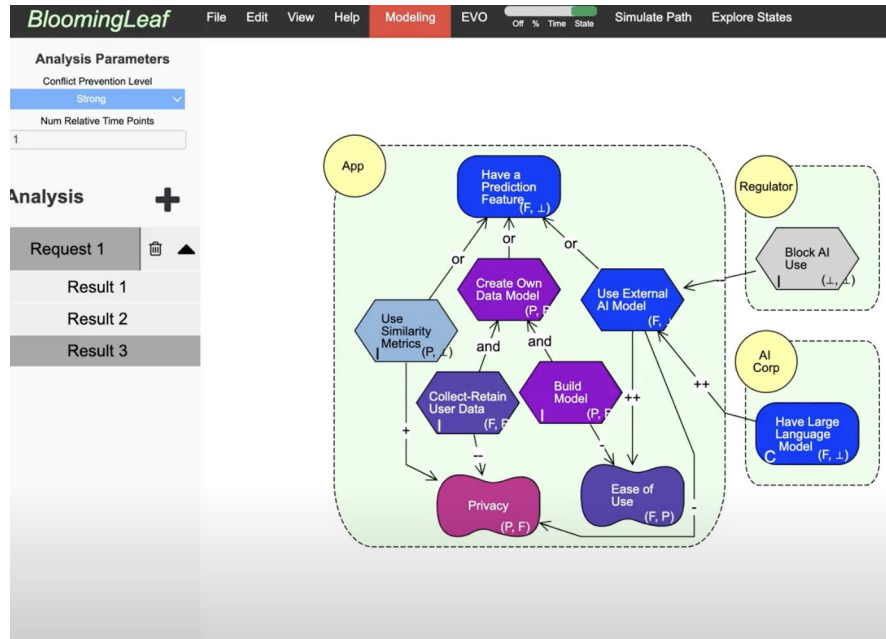
Why presence conditions? Example of the Predictive App



Simulation without presence conditions



Research Problem: Avoid nonsensical visualization after merging model segments from different time points



Misleading simulation visualization!

Our contribution

The screenshot displays the BloomingLeaf software interface. The top menu bar includes 'BloomingLeaf', 'File', 'Edit', 'View', 'Help', 'Analysis', 'EVO' (with 'Off' and 'On' options), 'View Assignments List', and 'Intermediate Values'. On the left, a legend identifies symbols: Goal (orange rounded rectangle), Task (green hexagon), Soft Goal (orange rounded rectangle), Resource (blue rectangle), and Actor (yellow circle). The main workspace shows a goal diagram for an 'App' actor. The diagram includes tasks like 'Have a Prediction Feature', 'Create Own Data Model', 'Use External AI Model', 'Use Similarity Metrics', 'Collect-Retain User Data', and 'Build Model'. It also features soft goals 'Privacy' and 'Ease of Use', and a 'Regulator' actor with the task 'Block AI Use'. A right-hand panel allows configuration of the 'Regulator' actor, showing its name, type (set to 'Actor'), a 'Hide Actor' checkbox, availability range (60 - 84), and a 'Flip Interval' field.

Incorporated presence conditions into the Evolving Intentions framework to avoid nonsensical visualization

Presences conditions are implemented as part of BloomingLeaf in the front-end actor

The screenshot displays the BloomingLeaf software interface. The top menu bar includes 'BloomingLeaf', 'File', 'Edit', 'View', 'Help', 'Analysis', 'EVO', 'View Assignments List', and 'Intermediate Values'. The main area shows a diagram with nodes like 'Feature', 'Create Own Data Model', 'Use External AI Model', 'Collect-Retain User Data', 'Build Model', 'Similarity Metrics', 'Privacy', and 'Ease of Use'. A callout box points to a 'Regulator' node with the text 'Intervals: [60, 84]'. To the right, an 'Actor Panel' is visible, showing 'Actor name: Regulator', 'Actor type: Actor', and 'Available: 60 - 84'. Another callout box points to the 'Actor Panel' with the text 'Intervals: [36, 84]'. The bottom left corner contains logos for GSUB, CAE, and Joints, along with release and copyright information.

Presence conditions specifications

The screenshot shows the BloomingLeaf software interface with a UML diagram. The diagram includes nodes for 'Feature', 'Regulator', 'Use', 'AI Corp', 'Have Large Language Model', 'Privacy', and 'Ease of Use'. Callout boxes provide the following specifications:

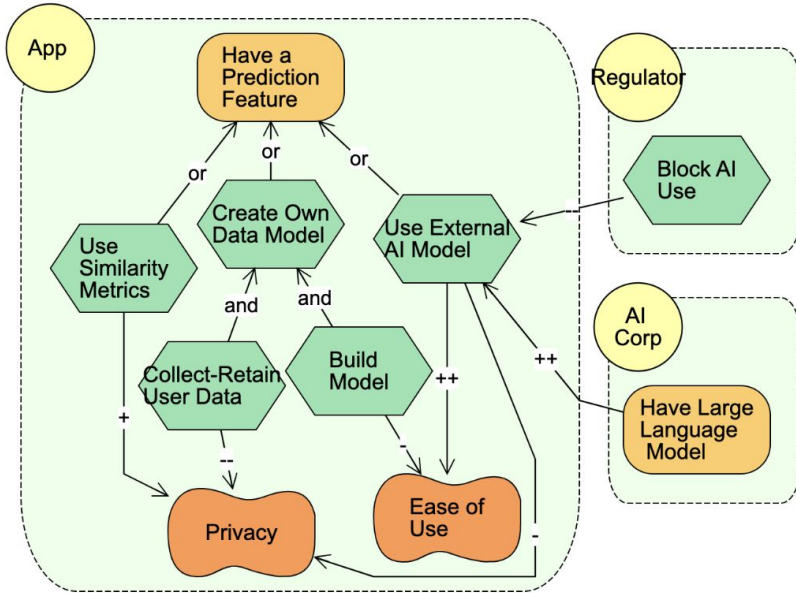
- Left Callout:**
Inclusion: [60, 84]
Exclusion: [0, 59]
- Right Callout:**
Inclusion: [36, 84] (User Perspective)
Exclusion: [0, 35] (System Specification)

Other interface elements include a menu bar (File, Edit, View, Help, Analysis, EVO), a status bar (Available: 60 - 84), and an Actor Panel.

Presence conditions were specified as **exclusion periods** to enable backward compatibility

Presence conditions specifications - Actor

actor $a \in A$: $\langle name, type, intention-set, exclusion-set (optional) \rangle$



Inclusion: [0, 84], 84 is **Maximum absolute time**
Exclusion: []

Inclusion: *User perspective*
Exclusion: *System specification*

Presences conditions are implemented as part of BloomingLeaf in the front-end intention

The screenshot displays the BloomingLeaf software interface. On the left, a legend identifies symbols for Goal (orange rounded rectangle), Task (green hexagon), Soft Goal (orange rounded rectangle), Resource (blue rectangle), and Actor (yellow circle). The main workspace shows a task diagram with nodes: 'App' (Actor), 'Have a Prediction Feature' (Task), 'Create Own Data Model' (Task), 'Use External AI Model' (Task), 'Use Similarity Metrics' (Task), 'Collect-Retain User Data' (Task), 'Build Model' (Task), 'Privacy' (Soft Goal), 'Ea Us' (Soft Goal), 'Regulator' (Actor), 'Block AI Use' (Task), 'AI Corp' (Actor), and 'Have Large Language Model' (Task). A red box highlights the 'Regulator' actor and the 'Block AI Use' task. A grey box at the bottom of the diagram contains the text 'Intervals: [36, 84]'. On the right, the 'Intention Panel' is visible, showing 'Node Name: Use External AI Model', 'Initial Satisfaction Value: None (1, 1)', and 'Function Type: No Function'. Below this, an 'Intervals' slider is shown with a blue dot at 36 and another at 84, with the text 'Available: 36 - 84' and a 'Flip interval' button.

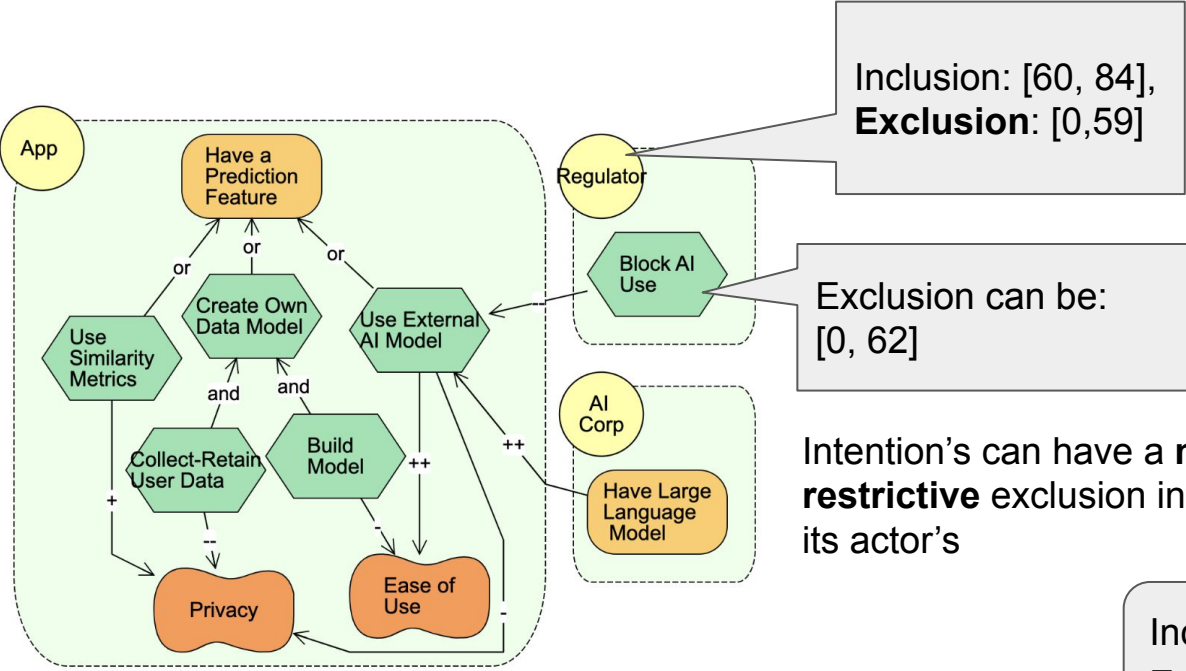
Release 2.0
Copyright 2015-2021,
Grubb Lab, Smith College
& University of Toronto.
All rights reserved.

Powered by
Copyright 2014-2021,
client I/O. All rights reserved.
JointJS: an HTML 5 diagramming
component.
<http://jointjs.com>

Intention Panel

Presence conditions specifications - Intention

intention $g \in G$: $\langle name, type, intention-set, exclusion-set (optional) \rangle$



Inclusion: [60, 84],
Exclusion: [0,59]

Exclusion can be:
 [0, 62]

Intention's can have a **more restrictive** exclusion interval than its actor's

Inclusion: *User perspective*
 Exclusion: *System specification*

Model-level view of presence conditions assignment

Absolute and Relative Assignments ✕

Max Absolute Time

Absolute Time Points
e.g. 5 8 22

Relative Intention Assignments +

Epoch Boundary Name 1	Relationship	Epoch Boundary Name 2
-----------------------	--------------	-----------------------

Absolute Intention Assignments

Epoch Boundary Name	Function	Assigned Time	Action
---------------------	----------	---------------	--------

Absolute Relationship Assignment

Link Type	Source Node name	Dest Node name	Assigned Time	Action
-----------	------------------	----------------	---------------	--------

Presence Condition Assignments +

Element	Type	Available Interval	
AI Corp	Actor	[36, 59]	
Regulator	Actor	[60, 84]	
App	Actor		
Use External AI Model	Task	[36, 84]	

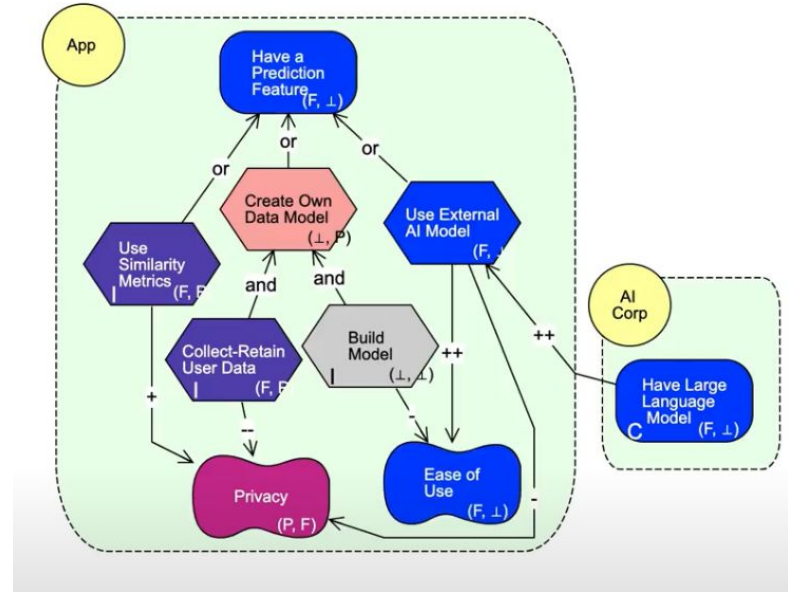
Help user keep track of every elements' presence conditions specification

Simulation with Presence Conditions

BloomingLeaf File Edit View Help **Modeling** EVO Of % Time State Simulate Path Explore States

Analysis Parameters
 Conflict Prevention Level: Strong
 Num Relative Time Points: 1

Analysis +
 Request 1
 Result 1
 Result 2

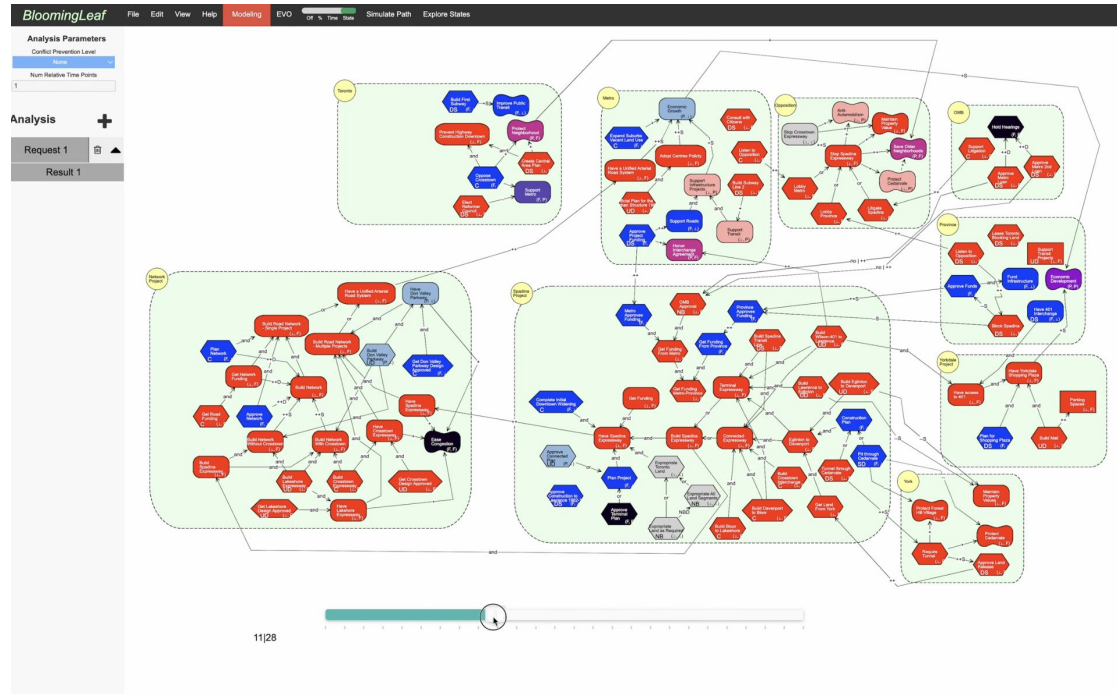


Has a better representation of when does certain actors come into play

Evaluation - Our approach is scalable, tested by a model over 100 intentions

Model of a large road
construction project*

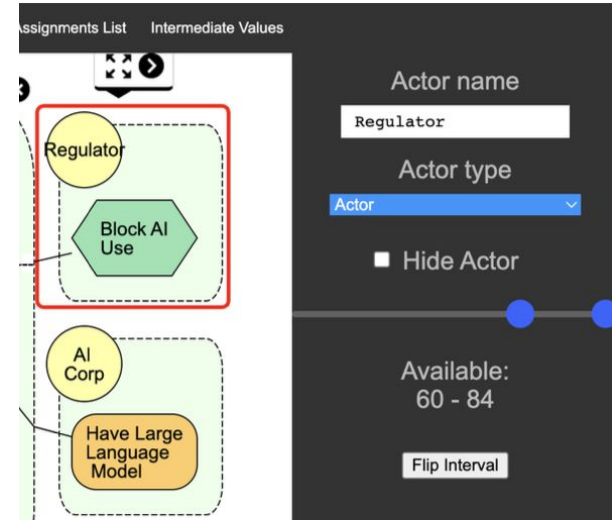
Can be improved by
allowing to select multiple
intentions at one time



* A.M. Grubb and M. Chechik. Reconstructing the Past: The Case of the Spadina Expressway. *Requirements Engineering (REJ)*, 25(2):253–272, 2020.

Model slicing

Using presence conditions as model slices allows us to empirically observe how stakeholders interpret large models.



Similar 'Hide' function in the analysis mode
Allow us to study how users explore large models

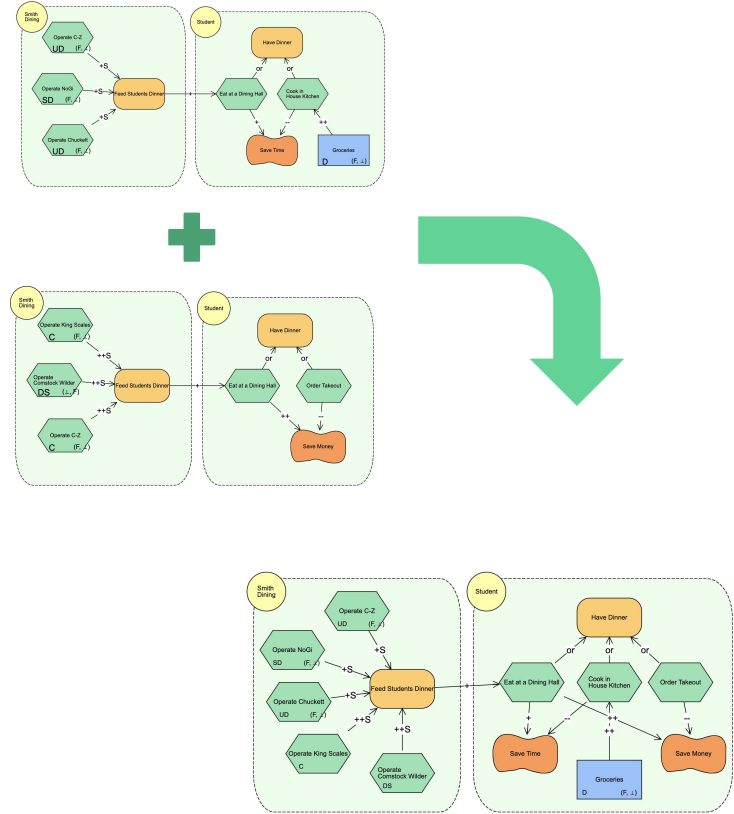
Ongoing Work & Future Work

Next steps:

Improve backend analysis to take presence conditions into account

Incorporate presence conditions with model merging and slicing

Generalizing our approach to other goal modeling languages



Related Work

Used for specifying software product lines in model-driven engineering

- Software product lines variability
- Domain Specific Languages (DSLs) for mapping feature models

Goal modeling:

- TimeURN

While we specify presence conditions as **temporal ranges**, not boolean conditions

Summary

We extended the Evolving Intentions framework to enable analysts to specify presence conditions

We integrated this extension in one of the goal modeling tools, BloomingLeaf

Acknowledgements:

We thank the members of our lab who helped in tool development and gave us feedback on this project: Venus Nguyen, Emily Kung, Karenna Kung, Thu Tran, and Caroline Zouloumian

This material is based upon work supported by the National Science Foundation under Award No. 2104732.