Exploring Next States and Alternative Paths in Goal Model Analysis

Boyue Hu, Alicia M. Grubb and Marsha Chechik

April 1, 2019

1 INTRODUCTION

GORE is concerned with the use of goals in various requirement engineering activities such as eliciting, elaborating, structuring, specifying, analyzing, negotiating, documenting, and modifying requirements and it has been advocated to help stakeholders make trade-off decisions in the early stages of project development by analyzing models that contain intentions (goals), system requirements and constraints in projects [1, 2, 3].

BloomingLeaf is a web-based goal modeling tool with automated formal analysis [4]. BloomingLeaf allows stakeholders to model changes in projects by assigning evolving functions to intentions in the model. When stakeholders specify questions about the project's evolution over time, the current analysis of the tool generates a single evolution path: a collection of simulated status of a predefined number of time points of the model [5]. However, one random possible evolution path might not be sufficient to understand the domain, and stakeholders may want to guide the generation of the path when one state is more favoured than another state. To better represent the users' problems and let them explore more possible solutions, we extend our analysis to allow users to customize their solution path by adding the functionality that supports stepping into any time point and exploring the solution space. This new functionality allows them to generate new solutions that better fit their projects.

In this abstract, we introduce our approach, *Explore Possible Next States*, to make BloomingLeaf's analysis interactive and help users better understand and evaluate potential project scenarios.

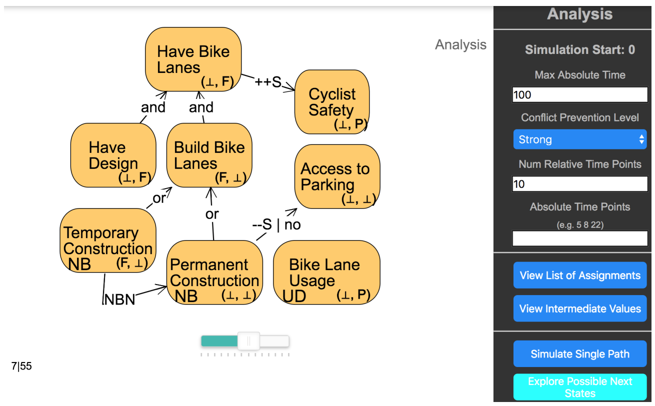

Figure 1: BloomingLeaf: Analysis View

2 BACKGROUND

Evaluation of Intentions.
When evaluation a goal model, we use a pair of information (s, d) (called evaluation pairs) where s (resp. d) represents the information for (resp. against) the satisfaction of the intention and we define three different values for the information: *fully satisfied*, *partially satisfied*, *no information* (resp. *fully denied*, *partially denied*, *no information*).
Evolving Intentions.
In BloomingLeaf, we describe how the value of any element in the model changes with evolving functions. These functions can be an ordered list of one or more *atomic* functions (i.e., Constant, Increase, Decrease or Stochastic) [5]. For example, the evaluation value of an intention with Increase function would gradually move towards *fully satisfied* and become less denied.

3 FRAMEWORK

Initially, the initial analysis of the tool allows the user to generate a single evolution path solution for their model. Then, as shown in Fig. 1, with the extended functionality, users are able to pick any time point from the single path solution and click on the Explore Possible Next States button. Then the tool displays all the possible next states in a new page.

Fig. 2 gives a screenshot of the exploring possible next states analysis

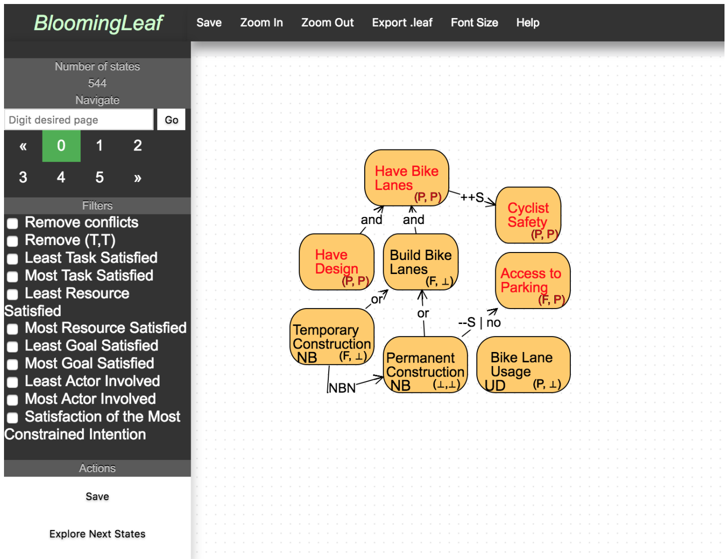

Figure 2: BloomingLeaf: Explore Next States View

page in BloomingLeaf. Users can then complete the following actions:

1. Explore all the possible states and select a preferred state. In Fig. 2, there are 544 possible next states listed by their indices, as shown above the navigation bar. Users are able to switch between states by selecting different indices and display selected states on the canvas.

2. Add the preferred state into the path and either (a) explore the possible states for the next state in the path, or (b) generate the remainder of the path with the selected state.

   (a) The button Explore Next States (see Fig. 2) allows the users to use the selected state on this page as the current state and further explore next possible states on the path incrementally until they reach the number of time points in the path.

   (b) The button Save allows users to save this selected state back to the original single path solution and complete the path and list it as a new solution as shown in Fig. 3.

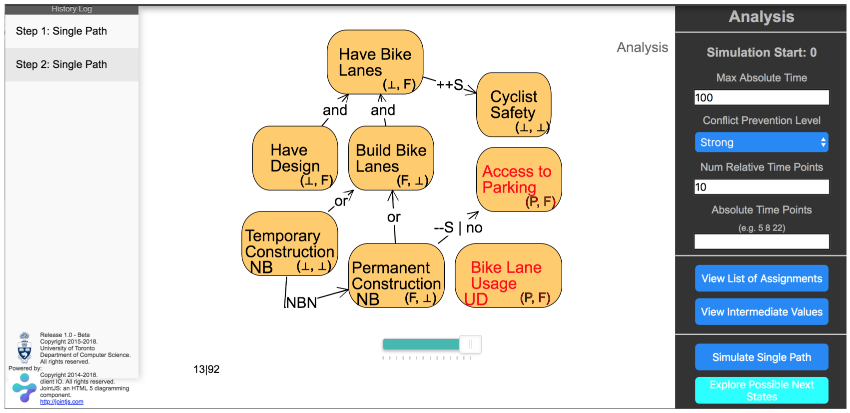

Figure 3: Screenshot of generated full path with selected state added.

4 ANALYSIS

Exploring Possible Next States is an extension to the current analysis algorithm to enable users to explore any time point in a single path solution (i.e., simulation).

The algorithm takes the goal model, the current time point and the partial time line generated by the single path analysis as inputs and it outputs all the possible next states. After the user clicks either Save or Explore Next States on the next states analysis page (see Fig. 2), the front-end of BloomingLeaf saves the selected state as the current time point in the path, and then passes the updated partial path (including the current time point) as input to the back-end.

Function transitions are important for analysis. Function transitions exist for evolving functions with two or more atomic functions. When analyzing the partial path, the algorithm records all function transitions in the evolving functions that have already happened before the current time point and if any function transition happens in the currently selected state. When a function transition has occurred before the current time point, it should impact all the time points afterwards. These transitions are encoded into constraints to differentiate whether a function transition is a possible state for the next time point. We prune the solution space based on the constraints in order to produce the correct result for the next state.

For example, imagine an intention is assigned two atomic functions, with an initial value of *fully denied*. In the first atomic function, the level of satisfaction increases until it reaches *fully satisfied*. In the second atomic function, the intention's value remains constant at *fully satisfied*. If the status of the intention already reached *fully satisfied* in the partial path, the only possible value for this intention in all future time points would be *fully satisfied*; thus, we limit the solution space for this intention to only include *fully satisfied*.

5 CONCLUSIONS

We introduced our approach to make BloomingLeaf's analysis interactive by allowing users to explore the entire possible solution space. By extending the current analysis algorithm, we implemented Explore Possible Next States to enable users to explore all the possibilities for every time point of their project and explore more alternatives of the project. By allowing the users to explore all the possible scenarios in their project at any time point in the simulation, stakeholders can better understand and evaluate potential project scenarios.

6 ACKNOWLEDGEMENTS

We would like to thank Marcel Serikawa for creating a proof of concept of the all path analysis.

References

[1] J. Horkoff, T. Li, F.-L. Li, M. Salnitri, E. Cardoso, P. Giorgini, J. Mylopoulos, and J. Pimentel, "Taking goal models downstream: A systematic roadmap," in *Proceedings of the IEEE 8th International Conference on Research Challenges in Information Science (RCIS'14)*, May 2014, pp. 1–12.

[2] J. Horkoff, F. B. Aydemir, E. Cardoso, T. Li, A. Maté, E. Paja, M. Salnitri, J. Mylopoulos, and P. Giorgini, "Goal-Oriented Requirements Engineering: A Systematic Literature Map," in *Proceedings of the 24th IEEE International Requirements Engineering Conference (RE'16)*, 2016, pp. 106–115.

[3] V. Lamsweerde, "GrowingLeaf: Supporting Requirements Evolution over Time," in *Proceedings of 9th International I* Workshop (iStar'16)*, 2016, pp. 31–36.

[4] A. M. Grubb and M. Chechik, “BloomingLeaf: A Formal Tool for Requirements Evolution over Time,” in *Proceedings of the 26th IEEE International Requirements Engineering Conference (RE’18): Posters & Tool Demos*, 2018.

[5] A. M. Grubb, “Modeling and analyzing the evolution of requirement over time using goal models,” Ph.D. dissertation, University of Toronto, 2018.