

Towards an Evaluation Visualization with Color

Megan H. Varnum, Kate M. B. Spencer, and Alicia M. Grubb

Department of Computer Science
Smith College, Northampton, MA, USA
{mvarnum, kspencer, amgrubb}@smith.edu

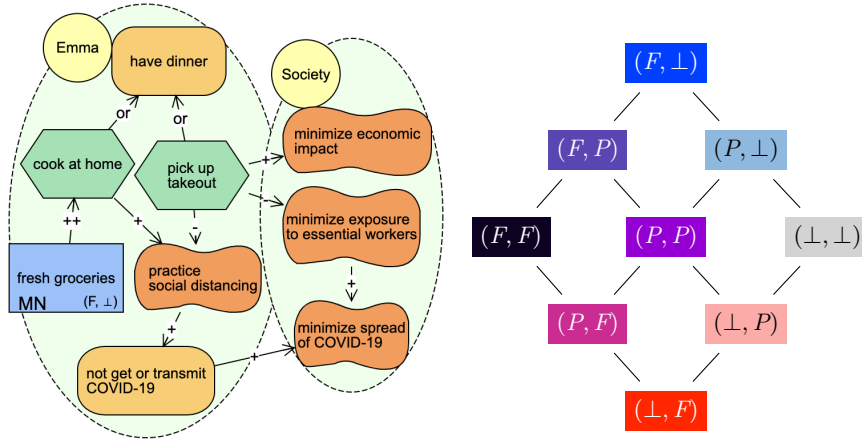
Abstract. Goal models help stakeholders understand project scenarios and make decisions. In prior work, we used Tropos evaluation semantics to allow for automated analysis over time; however, formal evaluation labels (e.g., (F, \perp)) are difficult for users to interpret across a large model. In this paper, we present our work towards understanding the extent to which using colors in goal modeling affects users' ability to make decisions. Specifically, we are interested in studying if coloring intentions with evaluation information allows for better comparisons of initial states and simulations of future paths. To address this question, we developed a color visualization extension to BloomingLeaf, a goal model analysis tool, where the color of each node is changed based on either the initial evaluations or the resulting analysis over time. This then allows us to explore if and how color visualization assists the user with decision making. We present our implementation and initial evaluation of this extension.

1 Introduction

Goal modeling allows stakeholders to model and visualize their domain and analyze trade-offs [1]. Recent work extends goal modeling and analysis to allow for project scenarios with changing evaluations [5]. Users can run simulations to examine relationships over time, with tooling and analysis provided for both iStar [7] and Tropos [6]. We believe these tools have potential for problem-solving among teams of stakeholders. However, the reality of working with large models is that it is challenging to view trends and make decisions because of the large volume of data and technical nature of analysis results.

In our work, we aim to improve the interpretability of model evaluations and analysis. We investigate the use of color to convey information about the fulfillment of evaluations, both statically and as they change over time. Previous work in the iStar community looks at improving clarity of goal models through visualizations and making them more user-friendly [3, 10]. We present our work in the context of BloomingLeaf, which uses Tropos semantics, but the lessons learned have similar implications for the iStar language.

Running Example: Social Distancing Dinner. In the midst of the COVID-19 pandemic [9], Emma must decide how to acquire dinner for the week. She constructs a goal model to evaluate the trade-offs between picking up takeout or cooking her dinner at home (see Fig. 1(i)). She runs simulations for various scenarios on BloomingLeaf, one where minimize economic impact is prioritized (i.e.,



(i) Running Example: Social Distancing Dinner

(ii) A lattice of evidence pairs in Tropos with colored labels.

Fig. 1: Goal Model and Evaluation Lattice

satisfied over time) and another where `minimize spread of COVID-19` is prioritized. For each scenario, BloomingLeaf returns evaluation labels for all timepoints in a week, and Emma is unsure how to proceed. How can she easily know whether `cook at home` or `order takeout` is satisfied more often? What about the proportion of time that `practice social distancing` is partially denied or fully denied? While traversing the simulation path, Emma is unsure how to interpret the data and unclear about the optimal course of action.

Contributions. In this paper, we explore our research question: To what extent does applying colors to individual intentions in models and simulation paths improve model understanding and allow for better decision making? We present Evaluation Visualization Overlay (EVO), an extension to BloomingLeaf which colors the initial satisfaction values (i.e., evidence pairs) of each intention in a goal model and, after running analysis, overlays the proportion of time points that each intention holds each evaluation. With EVO, users can better visualize future states of their model, make comparisons between intentions, and understand trends in fulfillment over time.

The remainder of this paper is organized as follows. Sect. 2 introduces relevant background about evaluation labels and BloomingLeaf. Sect. 3 explores our visualization approach. Sect. 4 discusses our preliminary evaluation and compares our work with prior visualizations. We conclude in Sect. 5.

2 Background

BloomingLeaf is an online tool used to construct and analyze goal models using the Tropos language [4]. Goal models consist of *actors* and *intentions* (i.e., goals, soft goals, tasks, and resources). For example, in Fig. 1(i), Emma is an actor, `have dinner` is a goal, `practice social distancing` is a soft goal, and `cook at home` is a task.

In BloomingLeaf, when EVO is not enabled, elements are solely colored based on their intention type. For instance, `fresh groceries` in Fig. 1(i) is colored light blue because it is a resource. These colors, combined with different shapes, allow users to distinguish intention types. In Tropos, each intention can be assigned a qualitative evaluation label (i.e., *evidence pair*), which is a pair (s, d) where $s \in \{F, P, \perp\}$ is the level of evidence for and $d \in \{F, P, \perp\}$ is the level of evidence against the fulfillment of an intention g [4]. F [resp. P] means there is full [resp. partial] evidence for or against the fulfillment of g , while \perp represents null evidence. Fig. 1(ii) gives a lattice of evidence pairs, with all possible combinations of evidence for and against. Initially, an intention can have one of five evaluation labels: *(Fully) Satisfied* (F, \perp) , *Partially Satisfied* (P, \perp) , *Partially Denied* (\perp, P) , *(Fully) Denied* (\perp, F) , and *None* (\perp, \perp) . Propagation-based analysis techniques may result in intentions being assigned one of four conflicting evaluation labels: (F, F) , (F, P) , (P, F) , and (P, P) .

Interactions between intentions are expressed through *links*. Contribution links (e.g., +, -, ++, -) indicate an element’s influence on another. For example, in Fig. 1(i), in order for `cook at home` to be satisfied, `fresh groceries` must also be satisfied, indicated by a ++, while `cook at home` helps `practice social distancing`, indicated by a +. Decomposition links (*and/or*) decompose an intention into child goals. An intention with an *and* [resp. *or*] decomposition requires all [resp. only one] of its children to be satisfied. *Functions* indicate an intention’s evolution. For example, Fig. 1(i) assigns `fresh groceries` a MONOTONIC NEGATIVE function (MN) because Emma’s groceries will expire throughout the week. After creating a model of intentions, links, initial values, and functions, users can analyze their model over time by simulating single paths (not shown). These are sequences of states consisting of evidence pairs for each element in the model over an ordered set of time points [5]. Users can step through these time points to see the changing evidence pairs of each intention, based on its assigned function. Unassigned intentions receive evidence pairs via propagation and the STOCHASTIC function. In Fig. 1(i), simulating a single path helps Emma see how the satisfaction of `fresh groceries` impacts the evaluation of `not get or transmit COVID-19` (not shown).

3 Results

The goal of Evaluation Visualization Overlay (EVO) is to visualize both the initial state and path analysis of a model over time. Each evidence pair is assigned a color, with bluer shades closer to satisfied and redder shades closer to denied, as illustrated in the background of Fig. 1(ii). Thus, (F, \perp) and (P, \perp) are shades of blue and (\perp, F) and (\perp, P) are shades of red. Conflicting evaluations with evidence for both satisfaction and denial are shades of purple, with (F, P) [resp. (P, F)] assigned shades of purple closer to blue [resp. red]. These colors are used both in modeling and analysis mode when EVO is enabled.

Implementation. Working within the architecture of BloomingLeaf, we implemented EVO as an on-off toggle option in the top toolbar, with its mechanics implemented in the JavaScript front-end. For the visualization of a single path

in analysis mode, BloomingLeaf’s back-end provides the necessary data, which is then stored in an encapsulating object. This object contains an array that corresponds to the evidence pairs of each intention at each time point.

EVO while Modeling. Activating EVO in the modeling mode changes the color of each intention to correspond to its user-set initial satisfaction value. This provides an overall visualization of the model’s initial state, allowing users to understand the color scheme and initial evaluations.

In our running example, Emma first prioritizes `minimize spread of COVID-19`, assigning it as (F, \perp) with a `CONSTANT` function. Knowing her groceries deplete over time, she sets the resource `fresh groceries` to (F, \perp) with a `MONOTONIC NEGATIVE` function. Interested in the initial state of the model, she activates EVO. She sees the two intentions above are blue (not shown), while the rest remain their original color. Because the majority of her model doesn’t change, Emma knows she cannot answer her dinner question with only the model’s initial state.

EVO while Analyzing the Resulting Model. After running a single-path simulation in analysis mode, activating EVO displays colored stripes on each element. The width of each colored stripe corresponds to the percentage of the path that the intention is assigned a given evidence pair, ordered left to right from most to least satisfied. Thus, the user can gauge the evaluations of intentions over time without needing to walk through the simulation state by state.

In our example, Emma, simulates a path in analysis mode and sees the evidence pairs in the corners of the intentions while stepping through time points. However, she is unable to remember data from each previous time point and so finds it impossible to make sense of the evidence pairs at once. She then turns on EVO to see how prioritizing the satisfaction of `minimize spread of COVID-19` impacts her dinner decision (see Fig. 2). While there are variations in the evaluations, Emma sees that `cook at home` is more satisfied than `order takeout`, and that it is never denied. This indicates that to minimize the transmission of COVID-19 in her life she should choose to cook at home. Emma also observes that `not get or transmit COVID-19` and `practice social distancing` are mostly satisfied, but notices that `minimize economic impact` has the majority of evidence towards denied.

In a second scenario, Emma wants to reduce job loss and is concerned about her local economy. She clears the model and sets `minimize economic impact` to fully satisfied with a `CONSTANT` function. Starting with EVO off, she is able to discern that while `order takeout` is fully satisfied, `minimize spread` is denied for several time points. However, toggling EVO reveals the whole picture of her results (see Fig. 2). While `order takeout` and `have dinner` are overwhelmingly satisfied, the other intentions have mixed results, and many are denied. Given this information, Emma understands the consequences of interacting with others and decides to cook at home to keep her and others the most safe.

EVO helps Emma understand trends in the model and answer her trade-off question: Which dinner option is fully or partially satisfied for the greatest proportion of time points under her chosen constraints? She can also draw conclusions about the unpredictability of her scenario with EVO by seeing discrepancies among multiple single paths.

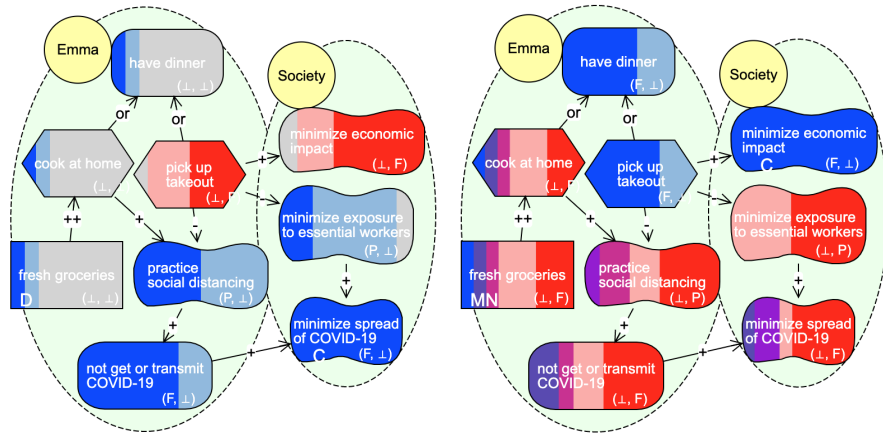


Fig. 2: Two single-path simulations for the running example. The left prioritizes minimize spread of COVID-19, and the result favors Emma cooking at home. The right prioritizes minimize economic impact, and the result favors ordering takeout.

4 Discussion and Related Work

To explore EVO’s versatility, we took nine models ranging in size from 14 to 121 intentions, some more deterministic and others more stochastic, and analyzed them with EVO enabled. For each model, we considered what information was clear before activating EVO and what we could not understand without it. We consolidated our observations into benefits of EVO and areas of improvement.

Benefits. The toggle operation makes the user interface easy to use. In the modeling mode, EVO shows initial evaluations to visualize the initial state of the model. The colors are intuitive, with bluer shades equating to more satisfied and redder shades to more denied. In analysis mode, EVO provides the user with the “big picture” of their model by visualizing the proportion of time points the intentions hold each evaluation. Since the order of colors is consistent, it’s easy to compare evaluation proportions across different intentions and simulation paths.

Improvements. The chosen color palette is not optimal for colorblind users. EVO can be improved by enabling a colorblind mode. An intention that starts (F, \perp) and evolves to (\perp, F) has the same visualization as the reverse, because the colors are always in the same order. An additional view showing changes in order would differentiate these two cases. EVO can also be improved to visualize any partial point of interest after simulating a single path. EVO is limited in the modeling view; it’s only useful for select models with well-defined initial states.

Plan for Validation. We have confidence EVO benefits users in drawing conclusions from goal models, given the distinct differences observed between using it and not; however, we only provide an initial evaluation. To quantify and validate the functionality of EVO, we must test it among a broader range of users.

We plan to conduct a study comparing BloomingLeaf’s usability in decision making with and without EVO by providing different models to two randomized

groups of students trained in goal modeling. One group’s tool will have EVO enabled, while the other group will only use the simulated path. The students will then answer questions about optimal decisions and evaluations over time.

Related Work. Aprajita proposed using heatmaps to depict overall satisfaction (see Fig. 11 of the associated thesis [2]). In both Aprajita’s work and this paper, we address the difficulty of allowing the user to visualize trends in time-based goal models [2]. Our research differs in that Aprajita’s heatmaps are separate from the goal model, whereas our work integrates the color visualization and the model, allowing users to see both at the same time. Aprajita’s system of red to green (for quantitative evaluations) does not apply in our work; thus, we develop our two axis color scheme (see Fig. 1(ii)).

5 Summary and Future Work

EVO visualizes evaluation trends in single-path simulations as well as models’ initial states, which allows for easy comparison across intentions within models. This work could have implications for decision making in other iStar tools [8].

Before drawing conclusions to our research question, a more thorough analysis across a broad range of potential users is required. Additionally, future work ensures our color palette is appropriately clear for all users. We are also developing a variation of EVO that visualizes the evaluations at every time point over a single path. This variation allows users to see changes of intentions over time instead of their unordered overall evaluation.

Acknowledgments. Varnum was supported by a Smith College STRIDE Scholarship.

References

1. D. Amyot et al. Evaluating Goal Models Within the Goal-Oriented Requirement Language. *International Journal of Intelligent Systems*, 25(8):841–877, 2010.
2. Aprajita. TimedGRL: Specifying Goal Models Over Time. Master’s thesis, McGill University, 2017.
3. R. F. de Oliveira et al. A Critical View Over iStar Visual Constructs. In *Proc. of iStar’19*, 2019.
4. P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani. Formal Reasoning Techniques for Goal Models. *Journal on Data Semantics*, 1:1–20, 2003.
5. A. M. Grubb. *Evolving Intentions: Support for Modeling and Reasoning about Requirements that Change over Time*. PhD thesis, University of Toronto, 2019.
6. A. M. Grubb and M. Chechik. BloomingLeaf: A Formal Tool for Requirements Evolution over Time. In *Proc. of RE’18 Posters & Tool Demos*, 2018.
7. A. M. Grubb, G. Song, and M. Chechik. GrowingLeaf: Supporting Requirements Evolution over Time. In *Proc. of iStar’16*, pages 31–36, 2016.
8. T. Li, A. M. Grubb, and J. Horkoff. Understanding Challenges and Tradeoffs in iStar Tool Development. In *Proc. of iStar’16*, pages 49–54, 2016.
9. WHO. WHO Statement regarding cluster of pneumonia cases in Wuhan, China. <https://www.who.int/china/news/detail/09-01-2020-who-statement-regarding-cluster-of-pneumonia-cases-in-wuhan-china>, Jan. 9 2020.
10. A. Yasin and L. Liu. Recent Studies on i*: A Survey. In *Proc. of iStar’17*, pages 79–84, 2017.