BOARD-AI: A Goal Recognition-Based Objective-Aware Modeling Interface for Systems Engineering

Alexandre Albore,[1] **Sandra Castellanos-Paez,** [2] **Nicolas Hili** [2]

[1] ONERA/DTIS, University of Toulouse, 31055 Toulouse CEDEX 4, France
[2] University Grenoble Alpes, CNRS, LIG, 38000 Grenoble, France
alexandre.albore@onera.fr, sandra.castellanos@univ-grenoble-alpes.fr, nicolas.hili@univ-grenoble-alpes.fr

Abstract

Paper and pens remain the most commonly used tools by systems engineers to capture system models. However, digitizing models sketched on a whiteboard into Computer-Aided Systems Engineering (CASE) tools remains a difficult and error-prone activity that requires the knowledge of tool experts. This study presents a reactive whiteboard interface for model sketching. This online approach combines techniques from Automated Planning and Machine Learning to improve performance while not compromising explainability of the system's output. Our approach mainly relies on two main modules: (1) a trained neural network that separates upstream from the global recognition process, handwritten text from geometrical symbols, and (2) a goal recognition algorithm that models the sketching task as a planning problem to identify the final sketches that are deemed possible. The main benefits of BOARD-AI are its autonomy, i.e. it does not rely on any other interaction modalities (e.g., virtual keyboards), and its explainability, i.e. the outcomes of the modeling assistant are understandable since a plan provides a valid explanation of the system's suggestions. Finally, BOARD-AI usability was validated trough user evaluations of engineers, experts and non experts in software and/or system modeling design. The demo video is available at https://cloud.univ-grenoble-alpes.fr/s/9ifZZCqW7eWQEdy.

Introduction

The usage of symbolic AI techniques to aid systems engineers to design models in a freehand way using large multi-touch screens is a viable process that has shown to be usable as an Human-machine interface for Systems Engineering modeling (Castellanos-Paez et al. 2022).

BOARD-AI is a reactive modeling interface for natural sketching that allows engineers to capture system models using interactive whiteboards. The heart of our approach lies in the use of *goal recognition* (Shvo and McIlraith 2020; Keren, Gal, and Karpas 2014; Lesh and Etzioni 1995; Kautz, Allen et al. 1986) techniques to translate user's sketches into model elements (Albore and Hili 2020; Hili, Albore, and Baclet 2021). More precisely, a modeling assistant identifies the most probable model elements intended to be drawn by a user from an initial sketch, even if partial. The outcome is an ordered list of suggestions ordered by the probability that a complete model element corresponds to the user's intent.

The main benefit of relying on symbolic AI rather than on ML is *explainability*, i.e. the property of a system that provides an output that makes understandable to the human user the reasons of an algorithm's choice. This condition is needed by any process-directed tool that allows users to evaluate the criteria behind a choice to use the tool more efficiently (Rosenfeld and Richardson 2019). Not only the modeling assistant provides the user with a list of suggestions, but it also details the remaining steps to draw the suggested model elements completely.

Finally, we conducted an evaluation that helped us to understand how BOARD-AI supports and facilitates the work of system engineers, and whether an AI-based modeling environment is trusted and deemed usable by its users. We obtained very encouraging results about usability, and AI-assisted sketching (Castellanos-Paez et al. 2022).

Goal-Aware Natural Sketching

The architecture of BOARD-AI follows a client-server architecture where the back-end (in Python) is responsible for calling the different services for shape recognition. It consists of three main modules. The *handler* module makes the link with the interface. It starts a Web-socket server to communicate with the front-end. Shape recognition being incremental, the web-socket communication is used to update the interface every-time a more optimized plan is found. A *Classifier* module built on top of *numpy* distinguishes geometrical shapes from textual annotations. The *planner* module is responsible for translating sketched elements into PDDL. Models elements to be recognized are structured as JSON objects and stored in separate files. When the *planner* module starts, it first loads the different files and then translates each JSON object into PDDL problem templates.

On the usage of Goal Recognition

We use PDDL to formalize our drawing problem, describe the initial sketch, and describe the list of all the model elements deemed possible. This list constitutes the *goal library* for goal recognition. The PDDL domain definition contains the formalization of the drawing problem.

This formalization of the drawing problem and the goal library are generic and reused across different executions of

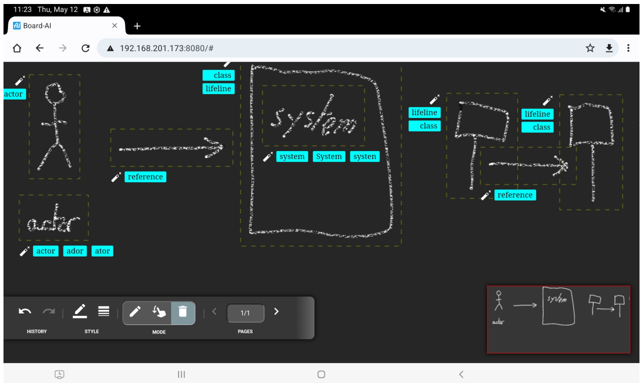

Figure 1: Screenshot of the BOARD-AI interface running on a Samsung Galaxy S6 Lite. Blue bubbles along with drawn sketches provide hints about the model elements to recognize.

the recognition process. Only the formalization of the initial sketch in PDDL is specific and is automatically generated from the graph obtained in the previous step.

Given the current state of the board and the goal library, we run the planner for the sketch being drawn by the user. We used the Fast Downward planning system (Helmert 2006), running the version of LAMA Planner (Richter and Westphal 2010) that participated in IPC 2011, and that has been integrated into Fast Downward's code. The anytime algorithm used in LAMA does not continue the weighted A* search once it finds a solution. Instead, it start a new weighted A*-based search from the initial state. The planner then outputs several plans, with increasing quality (measured in the number of actions in the plan). Each plan is an ordered list of possible matches between the sketch being drawn by the user and the goals denoting the different model elements that could be recognized. The set of possible matches is ordered based on the degree of confidence of the match regarding the element currently drawn. The degree depends on the *distance* (in the plan) between the current sketch and the possible goal, i.e., the number of steps that would remain to finish drawing the element completely.

Implementation details

Fig. 1 pictures the BOARD-AI main interface. The interface is developed using Web technologies (HTML, CSS, and JavaScript) so it can be used remotely and it can be run on any interactive pen display devices, ranging from tablets to large screens equipped with stylus. We adopted a minimalist style where the entirety of the screen can be used to draw a model so that users can fully focus on the sketching activity without being disturbed by an overloaded interface. The main area is an HTML5 Canvas for drawing model elements. A toolbar provides some useful features, such as undo/redo, page management, different edition modes (drawing, erasing, and selection), and two options to customize the thickness and the color of the drawing. The toolbar can be collapsed to maximize the drawing area. Besides these features, a chalk effect is applied to replicate the writing on a blackboard.

Two distinct interaction modalities are used to interact with the screen. Drawing is only possible using the stylus while touch gestures enable the user to navigate across the interface. Zooming in and out is achieved by pinching the screen. Single finger panning can also be used to navigate within the Canvas when it is scaled up. When the interface is scaled up, an outline at the bottom right of the screen shows the visible area.

When the user starts drawing, the recognition process is performed. Under the hood, the recognition engine consecutively identifies and characterizes the primitive shapes drawn by the user, invokes the classifier, and performs text or shape recognition according to the output of the classifier. Visual hints are given in the shape of bubbles accompanying the elements to recognize. Clicking on a visual hint results in converting the partial drawing into the suggested model element or text. If the partial drawing is updated by the user, the suggestions are updated as well.

Multiple shapes and/or textual annotations can be recognized at the same time. Hints to geometrical shapes are provided by the Fast Downward planning system (Helmert 2006) and are updated according to the *anytime* algorithm used. A limit of the three best suggestions is kept and displayed along with the partial drawing. Recognizing textual annotation is done using third-party Optical Character Recognition (OCR) engines.

References

Albore, A.; and Hili, N. 2020. From Informal Sketches to System Engineering Models using AI Plan Recognition: Opportunities and Challenges. In *AAAI 2020 Spring Symposium Series*.

Castellanos-Paez, S.; Hili, N.; Albore, A.; and Pérez-Sanagustin, M. 2022. BOARD-AI: A goal-aware modeling interface for systems engineering, combining machine learning and plan recognition. *Frontiers in Physics*, 10.

Helmert, M. 2006. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26: 191–246.

Hili, N.; Albore, A.; and Baclet, J. 2021. From Informal Sketches to Systems Engineering Models Using AI Plan Recognition. In Lawless, W. F.; Mittu, R.; A, S. D.; Shortell, T.; and McDermott, T., eds., *Systems Engineering and Artificial Intelligence*, chapter 22, 451–469. Springer.

Kautz, H. A.; Allen, J. F.; et al. 1986. Generalized plan recognition. In *Proceeding of AAAI*, volume 86, 5. Philadelphia, PA.

Keren, S.; Gal, A.; and Karpas, E. 2014. Goal recognition design. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 24, 154–162.

Lesh, N.; and Etzioni, O. 1995. A sound and fast goal recognizer. In *Proceeding of the International Joint Conference on Artificial Intelligence*, volume 95, 1704–1710.

Richter, S.; and Westphal, M. 2010. The LAMA planner: Guiding cost-based anytime planning with landmarks. *Journal of Artificial Intelligence Research*, 39: 127–177.

Rosenfeld, A.; and Richardson, A. 2019. Explainability in human–agent systems. *Autonomous Agents and Multi-Agent Systems*, 33(6): 673–705.

Shvo, M.; and McIlraith, S. A. 2020. Active goal recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 9957–9966.