Mixed-Initiative Approach to Collaboration in the Mathematical Domain

Nadya Belov and **Joshua Shaffer**
Department of Computer Science
Drexel University
Philadelphia, PA 19104
{belov, jbs36}@drexel.edu

Summary

Using smart-phones for ad-hoc mathematical collaboration poses multiple user interface challenges. In this paper, software agents are used to lessen the cognitive load through automatic line labeling. Researchers in the human-computer interaction community have attempted to alleviate the problem of general usability through user interface engineering conventions (Myers 1994). However, these engineering approaches can be improved upon through the application of mixed-initiative principles.

Background

Advances in computing are changing the fundamental modalities of both work and collaboration. No longer must one concern himself with what are increasingly becoming trivialities such as location and time. This becomes increasingly evident as autonomous forms of communication become economical, and are adopted. At the forefront of this iterative improvement are smart-phones, offering their users both more portability than conventional mobile computing devices (e.g., laptops), and nearly ubiquitous networking. Naturally, collaboration requires data input; however, with only a stylus and a small—usually non-Qwerty—keyboard, there are significant hindrances to their adoption.

Mixed-initiative is defined as the process of software-assisted user operation. This paper demonstrates that mixed-initiative principles can be applied to increase the viability of smart-phones. Furthermore, we show that this is especially evident when used as a tool for collaboration and artefact creation in a problem domain that requires ad-hoc collaboration. Herein, software agents are used to assist the user in artefact creation in two ways: (1) sensing the user's actions, and (2) assisting in said actions. Through (1) and (2), this paper shows that the software agent can work to lessen the complexity of the user's task.

The demands of mathematical collaboration cannot be met with a mere relay chat; the lexicon of mathematics is insufficient in explaining itself—in particular, to those who are just learning it. Instead, figures and formulæ are needed to augment what without would be an avalanche of jargon.

One may wish to engage in ad-hoc mathematical collaboration using smart-phones, working anywhere at anytime over the Internet. Presupposing that there is an intuitive method for inputting domain specific knowledge, the whiteboard emerges as viable collaboration metaphor for such communication. Making the presupposition true, that inputting domain specific knowledge must be made easy, is the research locus of this paper.

Free-form and parametric drawings, by themselves, are inappropriate for mathematical collaboration; figures must be labeled. However, systems that require users to manually label figures are too taxing, given the limited input devices standard on most smart-phones. The trivial case of drawing and labeling a line requires the following steps:

1. Acquire possession of the stylus.
2. Draw the line.
3. Remove the stylus from contact with the screen.
4. Enter a label for line.

However, if the system is able to automatically label figures, the user would be required to carry out steps 1 & 2.

Herein, automatic figure labeling is our goal, and we meet it through the use of software agents. To our knowledge, no other methods exist for automatically labeling figures.

In our formulation of the figure labeling problem, all mathematical figures can be represented as a set of control points from which labels may be drawn. The problem of labeling a set of sites or points such that no two labels overlap has previously been studied, and has been proven to be *NP-Complete* (Wagner & Wolff 1995); thus, an optimal, algorithmic approach is not tractable.

Theory

Software agents have been used as pedagogical monitors and advisers in (Lester, Stone, & Stelling 1999). We outline an intelligent agent that monitors the user's activities on the whiteboard. The agent's task is to label each new shape the user draws. The placement of the label must be such that it does not overlap with any other label or figure in the drawing area. Labeling heuristics are employed by the agent and are chosen dynamically based on the agent's perception of the environment.

Labeling a mathematical figure requires first decomposing the figure into primitives (e.g., lines or points). For example, a triangle will be decomposed into three lines, two of which will have a blank or clear label.

Labeling an endpoint of a line is sufficient, however not always optimal. Because the user is at liberty to arbitrarily place figures adjacent to each other, this may lead to ambiguities. Likewise, using the mid-point is suitable; however, it may be ambiguous if figures are made to intersect. Thus, there are three ways to label a given line: endpoint, mid-point, and picking a location on the line that is suitably distant from any intersections or adjacent lines. Therefore, heuristics are employed for method selection.

After suitable label locations have been selected, this problem becomes isomorphic to the map-labeling problem (Neyer & Wagner 2000), which, as mentioned above, is known to be *NP-Complete*. Additionally, the heuristics given in Algorithms 1, 2, 3 and 4 select the locally optimal label orientation that minimizes the interactions with other figures or labels. The AVAIL function, referenced in all of the heuristics, returns the availability of a site for labeling. AVAIL is itself a heuristic, calculated in terms of the preexisting density of the neighboring region. The END-POINT heuristic is æsthetically optimal, as it follows traditional conventions. However, it is not necessarily possible to place a label at the endpoint of a line without conflict. In the case of conflict, the MID-POINT heuristic is used, in an attempt to place the label at the mid point of the line; however, that too may not be possible. There may be intersecting lines in the region of the midpoint, causing the resulting label to be un-associable with its corresponding line. The last heuristic is the least æsthetically desirable, but will guarantee a candidate point, if one exists. Once a valid candidate site is chosen for a label, the required orientation of the label is not necessarily clear. The fourth heuristic, given in Algorithm 4, determines the exact orientation of a label.

Algorithm 1 END-POINT(e_1, e_2)

Require: e_1 and e_2 are the endpoint coordinates of a line
Ensure: candidate is the location of the label
  $E \leftarrow \{e_1, e_2\}$
  if $(\exists i \in E \mid \text{AVAIL}(i) < \epsilon_e)$ **then**
    candidate $\leftarrow \arg\min_{e \in E}(\text{AVAIL}(e))$
  end if

Algorithm 2 MID-POINT(e_1, e_2)

  $mid_x \leftarrow \frac{1}{2}(e_1 + e_2)$
  if $\text{AVAIL}((mid_x, f(mid_x)) < \epsilon_h$ **then**
    candidate $\leftarrow (mid_x, f(mid_x))$
  end if

Algorithm 3 SAMPLING(e_1, e_2)

  for n times **do**
    x $\leftarrow$ RANDOM(e_1, e_2)
    y $\leftarrow f(x)$
    if $\text{AVAIL}(x, y) < epsilon_s$ **then**
      candidate $\leftarrow (x, y)$
    end if
  end for

Algorithm 4 QUADRANT-SELECTION(e_1, e_2)

  $Q = \{q_1, \cdots, q_4\}$
  for $i \in Q$ **do**
    if $\text{AVAIL}(i) < \epsilon_j$ **then**
      candidate $\leftarrow i$
      break
    else
      $Q \leftarrow Q - \{i\}$
    end if
  end for
  if $Q = \emptyset$ **then**
    candidate $\leftarrow \arg\max_{q \in Q}(\text{AVAIL}(q))$
  end if

Results and Future Work

Empirical results of our mixed-initiative labeling strategy (see Figure 1) show significant improvement in the time required to create an artifact. Furthermore, the map-labeling heuristics presented above showed effective æsthetic results. The diagrams are easy to understand because the labels were clearly associable with the appropriate shape. Extending

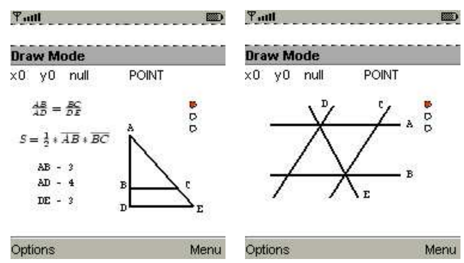

Figure 1: Automatic Labeling Results

this approach to learn individual user preferences is the next natural step in further reducing the user's cognitive load on mobile computing devices with limited input abilities. Towards this goal, the agent will utilize online learning techniques to learn individual user preferences.

Acknowledgments

The authors would like to thank Dr. Werner Krandick of Drexel University for proposing the automated line labeling as a research problem. This research was supported in part by NSF ITR-Grant REC-0325872.

References

Lester, J. C.; Stone, B. A.; and Stelling, G. D. 1999. User modeling and user-adapted interaction. *The Journal of Personalization Research* 9(1–2):1–44.

Myers, B. 1994. Challenges of HCI design and implementation. *Interactions* 1(1):73–83.

Neyer, G., and Wagner, F. 2000. Labeling downtown. In *Proceedings of the 4th Italian Conference on Algorithms and Complexity*, 113–124. Springer-Verlag.

Wagner, F., and Wolff, A. 1995. An efficient and effective approximation algorithm for the map labeling problem. In *Proceedings of the Third Annual European Symposium on Algorithms*, 420–433. Springer-Verlag.