

LangUI: Vector Drawing for Scientific Writing

Jipeng Wu

I. REQUIREMENT ENGINEERING

A. Motivation

Illustrations in research papers can be classified into 2 categories: data-based charts and concept-based illustrations. We don't usually worry about the first one since data-based charts can be generated automatically. The second one however could be painful. Those illustrations may not look like existing graphs or figures that diagramming tools already predefined. They usually try to use basic shapes and lines to convey the concept of a novel model. Existing tools like visio, Lucidcharts, metapost, LaTeXDraw or python support such creation.

But sometimes men are too lazy to drag components or write python/R/TikZ/metapost code.

I want to simply draw what it is in my mind and the software should understand my sketches, tidy up everything and produce some beautiful illustrations.

B. High level Objectives

The highest level goal is to develop a tool that understands users' handdrawing and produce illustrations for scientific writing.

More specifically, it should outperform visio, Lucidchart or code-based tools in efficiency and satisfaction. In terms of completeness and effectiveness, I would not expect full coverage of all features and functions that diagramming tools generally have. However, the LangUI program should at least be able to produce the following illustrations: "lines and curves", "arrows and texts", "basic shapes", "overlapping shapes".

Fig. 1. lines and curves

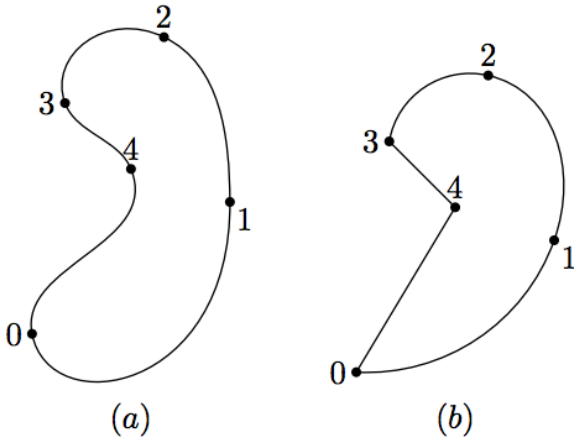


Fig. 2. arrows and texts

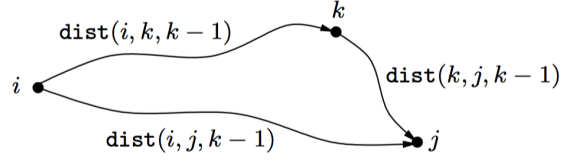


Fig. 3. basic shapes

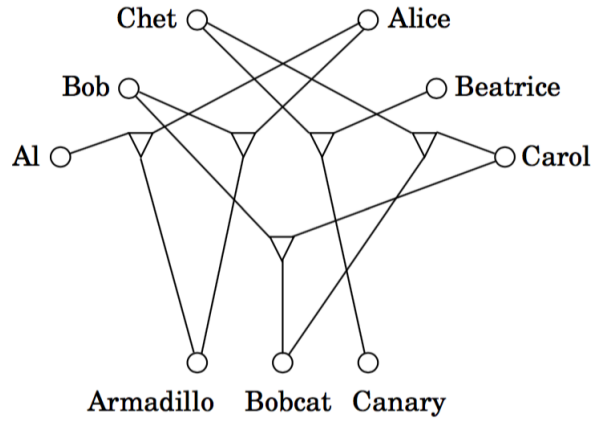
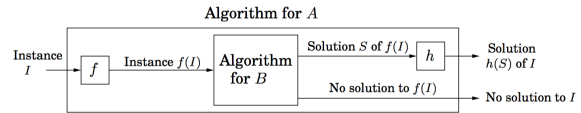


Fig. 4. overlapping shapes



C. Requirement Elicitation

II. RELATED WORK

Existing diagramming tools like visio, Lucidcharts provide a wide range of predefined components. Users select an item and then drag it to some place. When they finish drawing one item, the following item is usually different. So they have to search and click another item. This process is very tiring and especially frustrating when you forget to change the selected item.

Selecting and dragging mode will be a submode in Langui. I think it is still necessary to support this mode. So users are allowed to tweak their layouts after drawing the sketches.

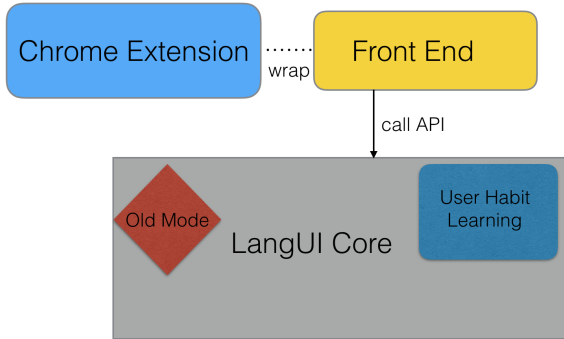
I always want to find a tool that tidies things up and magically makes it look professional. I haven't found anything like that. But I do find similar ideas

exploited in gesture understanding and Chinese stroke input method.

III. PROPOSED WORK

A. Architecture

Fig. 5. Architecture



LangUI is composed of 3 parts: a Google Chrome extension wrapper, a html5 front end and LangUI core.

When you write some reports or paper, you are probably using a keyboard and a mouse. So I wouldn't consider tablets even though they are better devices for hand drawing. To make this tool more available, cross-platform and lightweight, I decide to wrap it as Google Chrome Extension. The front end could be most likely written in html5. But it could change during implementation phase.

The LangUI core should provide the following features:

- 1) Understanding users' strokes: Since users draw in LangUI's front-end, the speed and direction of strokes can be tracked. LangUI can analyze these features to guess the users' intention. For example, I would prefer to draw a "C" or "6" very quickly to embody the concept of "circle", LangUI should differentiate these strokes from a slow and smooth "C"/"6"-like curve.
- 2) Tidying up the sketches to make them look like computer generated illustrations: LangUI should automatically generate smooth curves and straight lines. Automatic aligning is an optional feature.
- 3) Old mode: It is the selecting and dragging mode. It is optional, but I do want to finish it as it could be a crucial supplement for the proposed workflow.
- 4) Supervised learning: The LangUI learning submodule should track and learn users' preference on how to draw certain elements. The data could be stored on a server and linked to their social media account. But I think simply using cookie is also acceptable in this scenario. This feature is optional. If I don't have enough time, the default setting would be my personal preference.

B. Algorithms

IV. EVALUATION AND FUTURE WORK

A. Usability Analysis

B. Where I need to Rework or Improve

It's really hard to beat state-of-the-art diagramming tools in usability. To achieve this goal, I have to make sure LangUI's understanding of users' intention is extremely accurate. Any misunderstanding would probably ruin user experience. Therefore the major challenge is how to optimize the stroke understanding module.

C. Critical Future Features

V. BIBLIOGRAPHY

REFERENCES

- [1] S.W.Draper, *Practical Methods for Measuring Human-Computer Interaction*, Nov 2002. Link: <http://www.psy.gla.ac.uk/~steve/re-sources/uipm.pdf>.
- [2] John D. Hobby, *Metapost: A User's Manual*, May 21, 2014. Link: <https://www.tug.org/docs/metapost/mpman.pdf>.
- [3] T. Morales de Luna, *Useful Vector Graphic Tools for LATEX Users*, in *The PracTEX Journal*, 2010, No. 1. Link: <https://tug.org/pracjourn/2010-1/morales/morales.pdf>.