Beautifying sketching-based design tool content: issues and experiences

Beryl Plimmer[1] and John Grundy[1, 2]

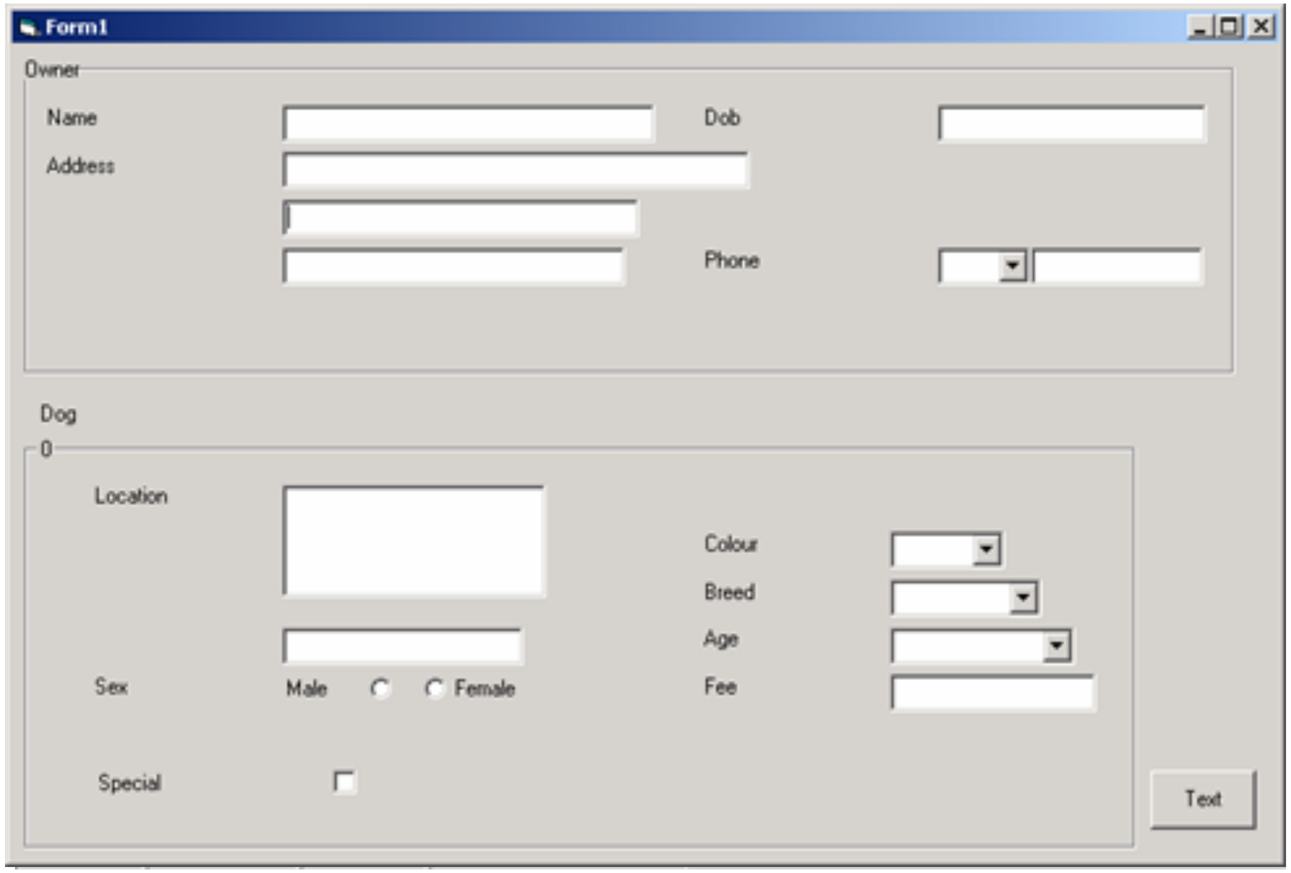
[1]Dept of Computer Science and [2]Dept of Electrical and Computer Engineering
University of Auckland, New Zealand

Outline

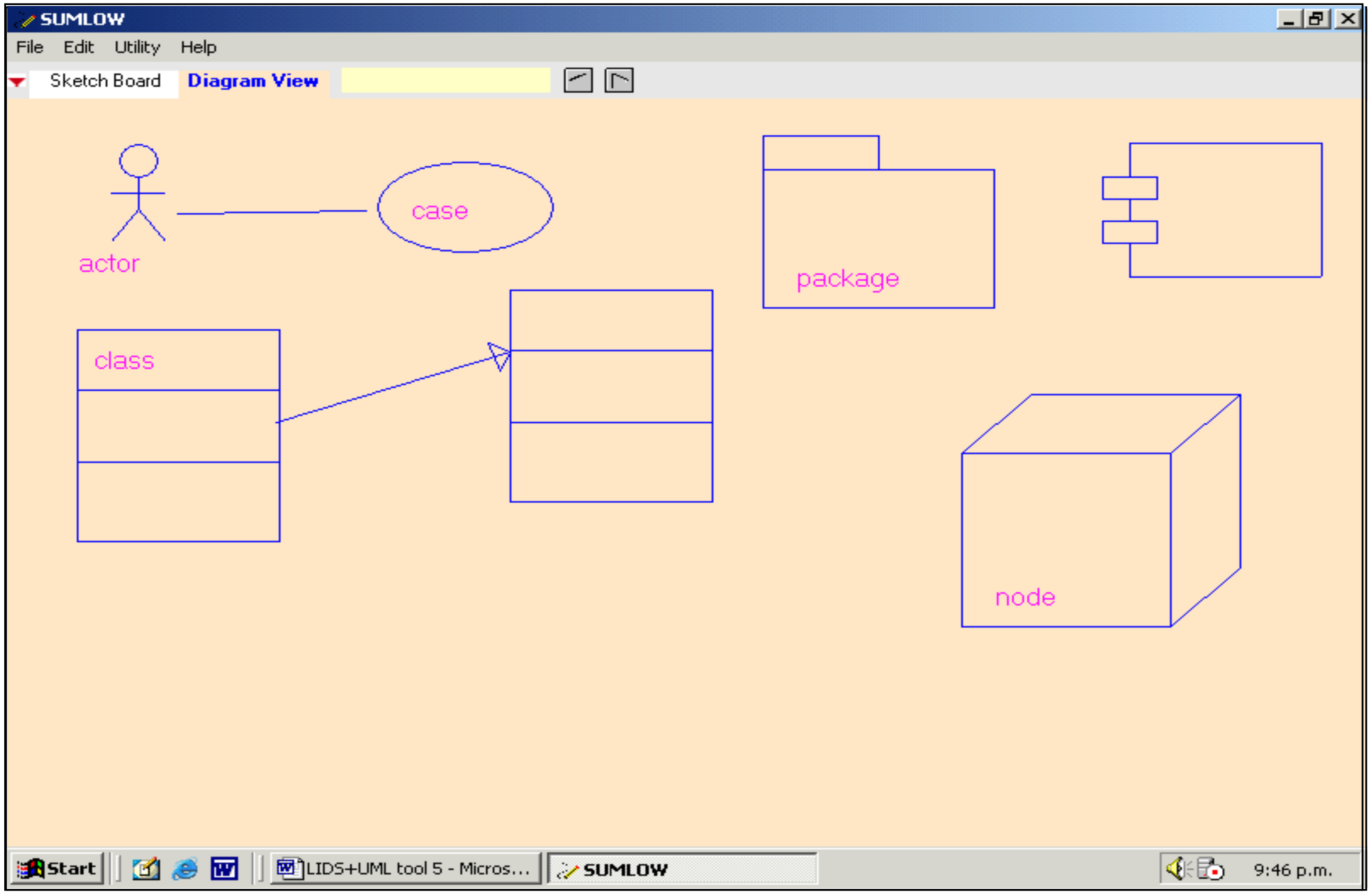
- Motivation
  - Why do we need to “beautify” sketched input?
- Requirements
  - What kinds of sketching tools are there?
  - Do they have different beautification needs?
- Examples
  - UML sketching and User Interface sketching
- Experiences
  - What works well? What doesn’t…
- Conclusions

Motivation

- An example:



Another Example…



Why do this?

- User adds elements, want modified e.g. sketch actor figure and want text edit area added
- User needs constraints enforced e.g. if put text box over another text box, move one of them or resize
- Want layout implemented e.g. UML sequence diagram
- When move/resize something, flow-on effects e.g. resize class icon => move enclosed text/connectors
- When formalise elements, need to apply standard formatting rules e.g. UI text label's font, size, colour, style, …
- Layout rules on formalising diagram content e.g. align radio buttons; auto-layout UML class diagram

How do we do this??

Sketch-time Beautifications:

- Recognize shape & modify appearance/location/size
- Auto-clustering related elements
- Element overlap removal
- Auto resize/move of related elements
- Alignment to grid

Formalisation-time Beautifications:

- Apply heuristics to sketched elements to convert to computer-rendered forms
- Apply grids, auto-layout algorithms
- Apply consistent formatting styles to elements

Different tools require different mix…

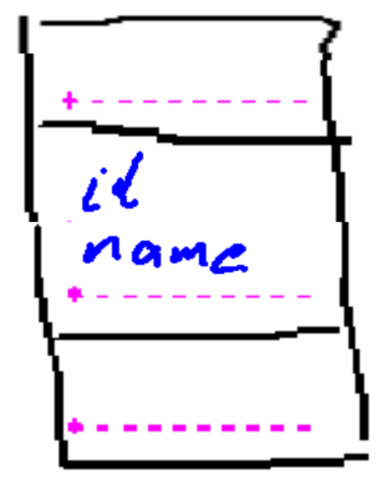
Examples: Draw and change

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

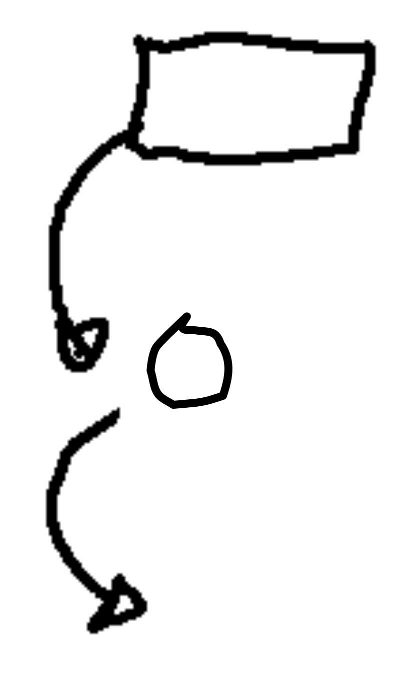
The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x

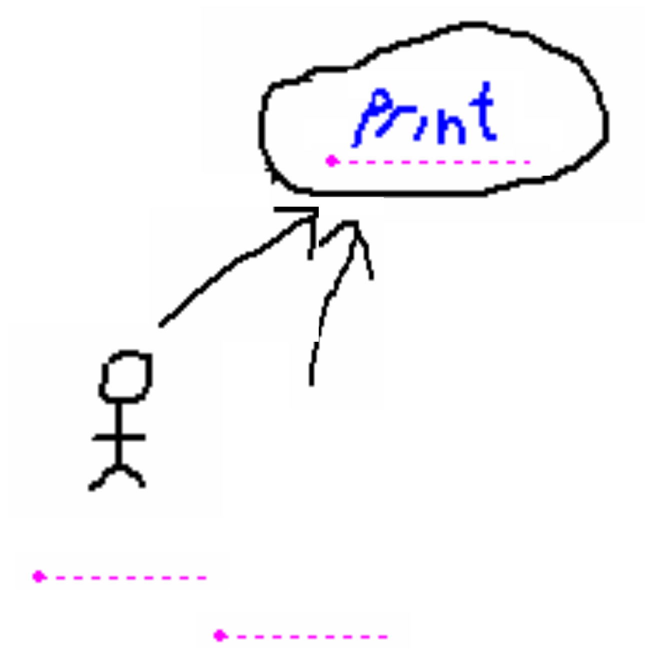
The image cannot be displayed. Your computer may not have



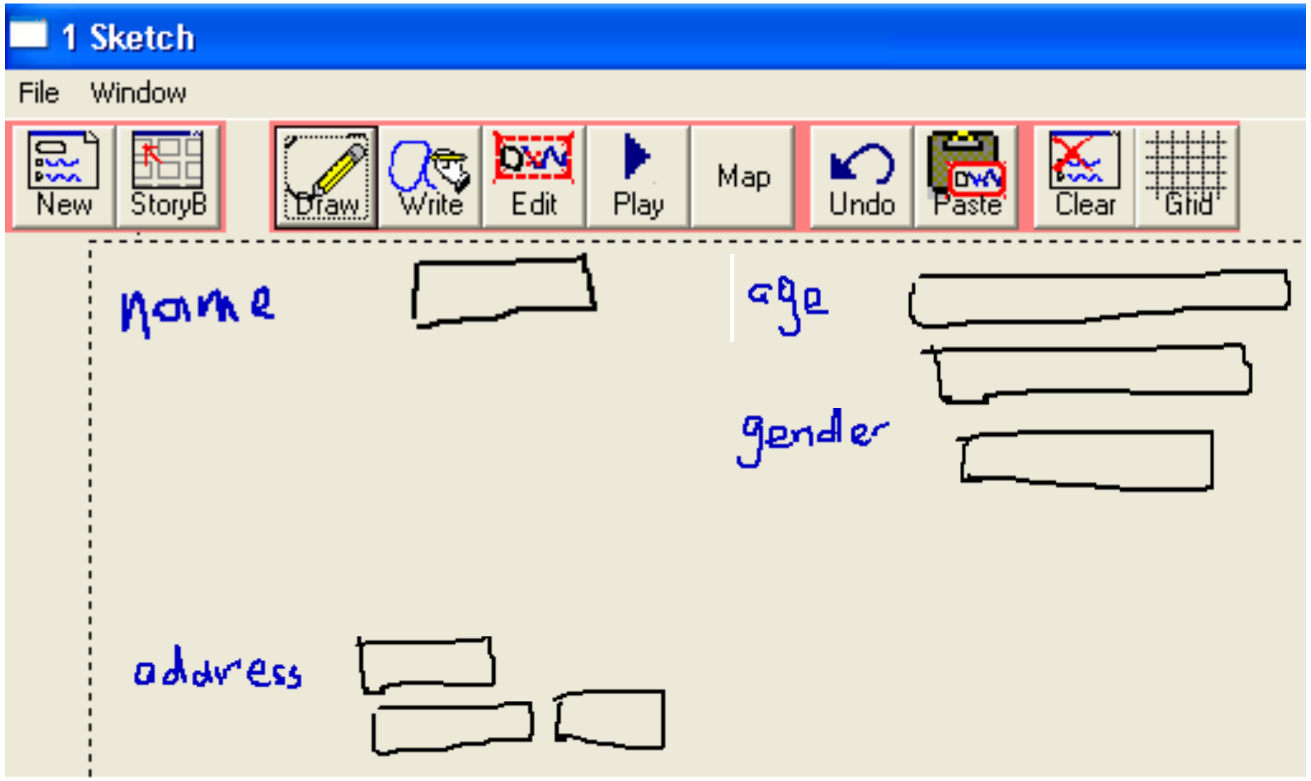
Remove and replace



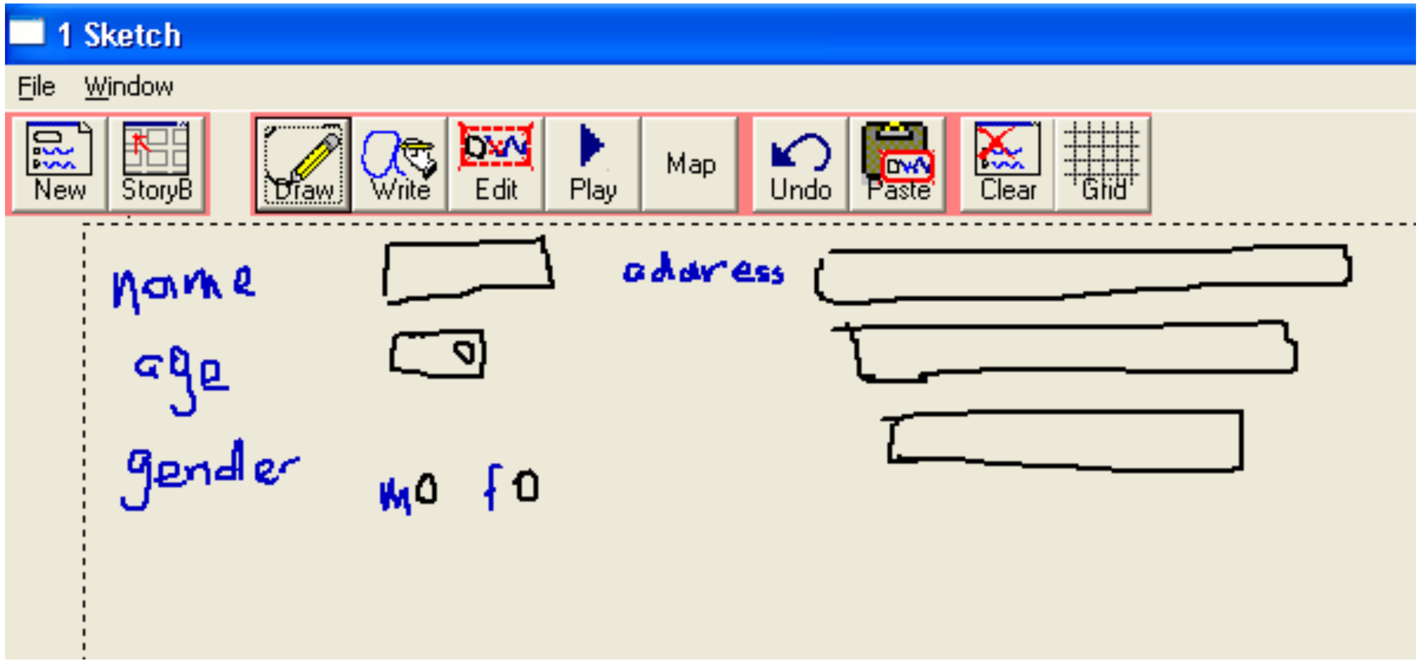
Move and resize

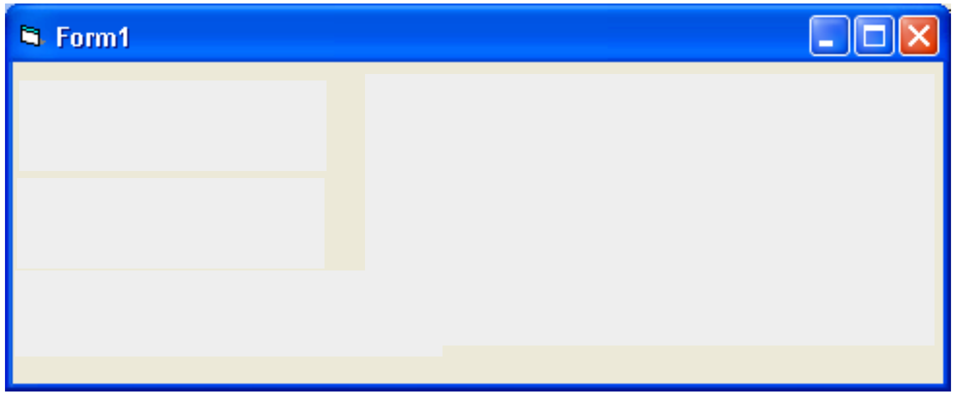


Move and resize group



Apply styles during formalise





Experiences

- In FreeForm:
  - Concrete layout & layout constraints v. important
  - Removing overlaps v. important; determining element groups important
  - Auto-placement, resize during sketch not always desired by user
  - Applying standard styles during formalisation v. important – user needs control over how these are done
- In SUMLOW:
  - Abstract design so user-defined layout/overlaps OK
  - Auto-adding text areas, auto-moving connectors v. important
  - Determining relevant groups during sketching necessary
  - Layout of sequence diagrams necessary during sketching; others auto-layout not needed during sketching
  - Can apply standard styles and layout algorithms during formalisation, but less necessary than in FreeForm

Conclusions & Future Research

- Beautification during sketching and formalisation important for usability of sketching-based UIs
- Different kind of design important – concrete vs abstract models; importance of layout/element interaction
- Users need adequate control over these however

- User configuration of beautification algorithms
- Implementation of beautification in different tools
- Taxonomy of diagramming-based tools to aid in development, including beautification strategies

References

- Plimmer, B. and Grundy, J.C. Beautifying sketching-based design tool content: issues and experiences, In Proceedings of the 2005 Australasian User Interfaces Conference, Jan 31-Feb 3, 2005, Newcastle, Australia, Conferences in Research and Practice in Information Technology, Vol. 40.
- Grundy, J.C. and Hosking, J.G. Supporting generic sketching-based input of diagrams in a domain-specific visual language meta-tool, In Proceedings of the 2007 IEEE/ACM International Conference on Software Engineering (ICSE'07), Minneapolis, USA, May 2007, IEEE CS Press.
- Chen, Q., Grundy, J.C., and Hosking, J.G. SUMLOW: Early Design-Stage Sketching of UML Diagrams on an E-whiteboard, Software – Practice and Experience, vol. 38 , no. 9, Wiley, July 2008, pp. 961-994.
- Blagojevic, R., Plimmer, B., Grundy, J.C. and Wang, Y. A Data Collection Tool for Sketched Diagrams, In Proceedings of the 5th EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling, Annecy, France, June 11-13, 2008.
- Blagojevic, R., Plimmer, B., Grundy, J.C. and Wang, Y, Development of techniques for sketched diagram recognition, In Proceedings of the 2008 IEEE Symposium on Visual Languages and Human-Centric Computing, 2008, pp 258-259.
- Patel, R., Plimmer, B., Grundy, J.C. and Ihaka, R. Ink Features for Diagram Recognition, 4th Eurographics Workshop on Sketch-Based Interfaces and Modeling, Riverside, California, August 2-3, 2007.