



*Supporting generic sketching-based
input of diagrams in a domain-
specific visual language meta-tool*

*John Grundy & John Hosking
University of Auckland, New Zealand*



Outline (back of napkin...)

- ✓ Motivation
- ✓ Current design tool interaction
- ✓ Sketching-based design
- ✓ Adding sketching to our meta-tool
- ✓ Key issues
- ✓ How we did it for Marama meta-tools
- ✓ Evaluation
- ✓ Future work
- ✓ Conclusions



Motivation

- ✓ CASE tools aren't used - Iivari etc
 - not until Model-Driven Development anyway... ☺
- ✓ Using pencil & paper the way most of us design software - architecture, UIs etc - Landay et al
- ✓ Several studies shown hand-drawn sketches facilitate "better" design - as well as being more "usable" for designers - Plimmer et al
- ✓ Very little support in current design tools for sketching-based input - despite e.g. TabletPC, stylus-based PDA popularity
- ✓ It's a fun idea to extend our meta-tool to add sketching support...
 - curiosity-driven research



Current design tool interaction

- ✓ Example: MaramaMTE
- ✓ Software architecture modelling, performance engineering tool
- ✓ Eclipse-based toolset produced with our Marama meta-tools
- ✓ Uses conventional interaction:
 - Mouse
 - Tool pallet
 - Drag-and-drop
 - Keyboard for text entry (including on Tablet PC)
 - Save/load diagram content o XML files
 - Share via CVS/SVN; diagram diff/merge etc



Example #1

The screenshot shows a modeling tool interface with a diagram and a properties table. The diagram features a central pink 'Server' box connected to a cyan 'WebUI' box, a green 'DB' box, and a light green 'CustomerService' box containing a pink 'Cust' box. The 'Shapes' and 'Connectors' panels on the left are circled in red, with a tooltip 'Create a ClientShape shape' pointing to 'ObjectShape'. The 'Properties' table at the bottom is also circled in red.

Property	Value
*Model Elements	
host	
kind	
Location	42, 37
name	Web UI
Size	142, 94
threads	

And so
on...



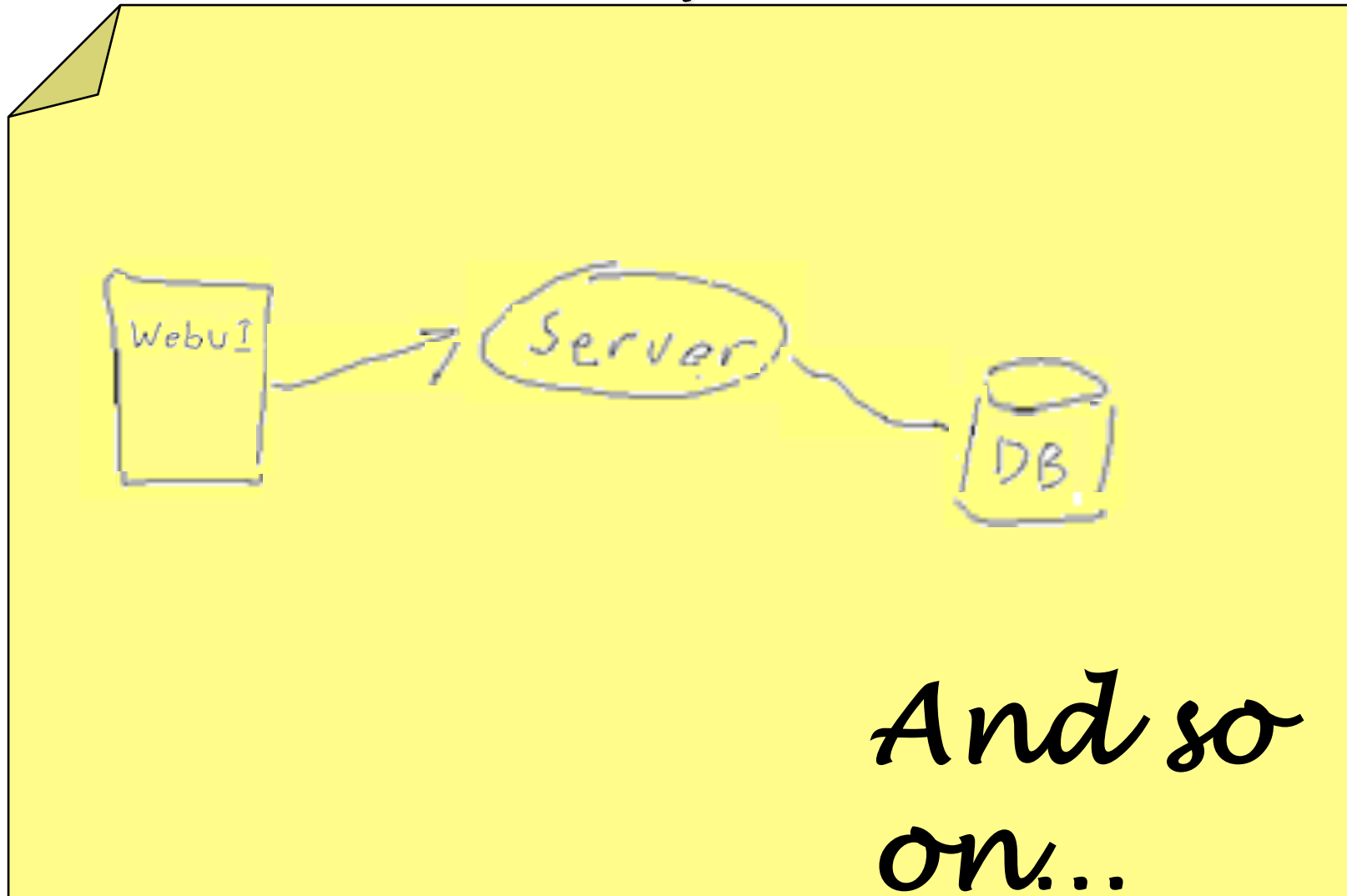
ICSE 2007 - Sketching-based
input & meta-tools

What do we REALLY do?

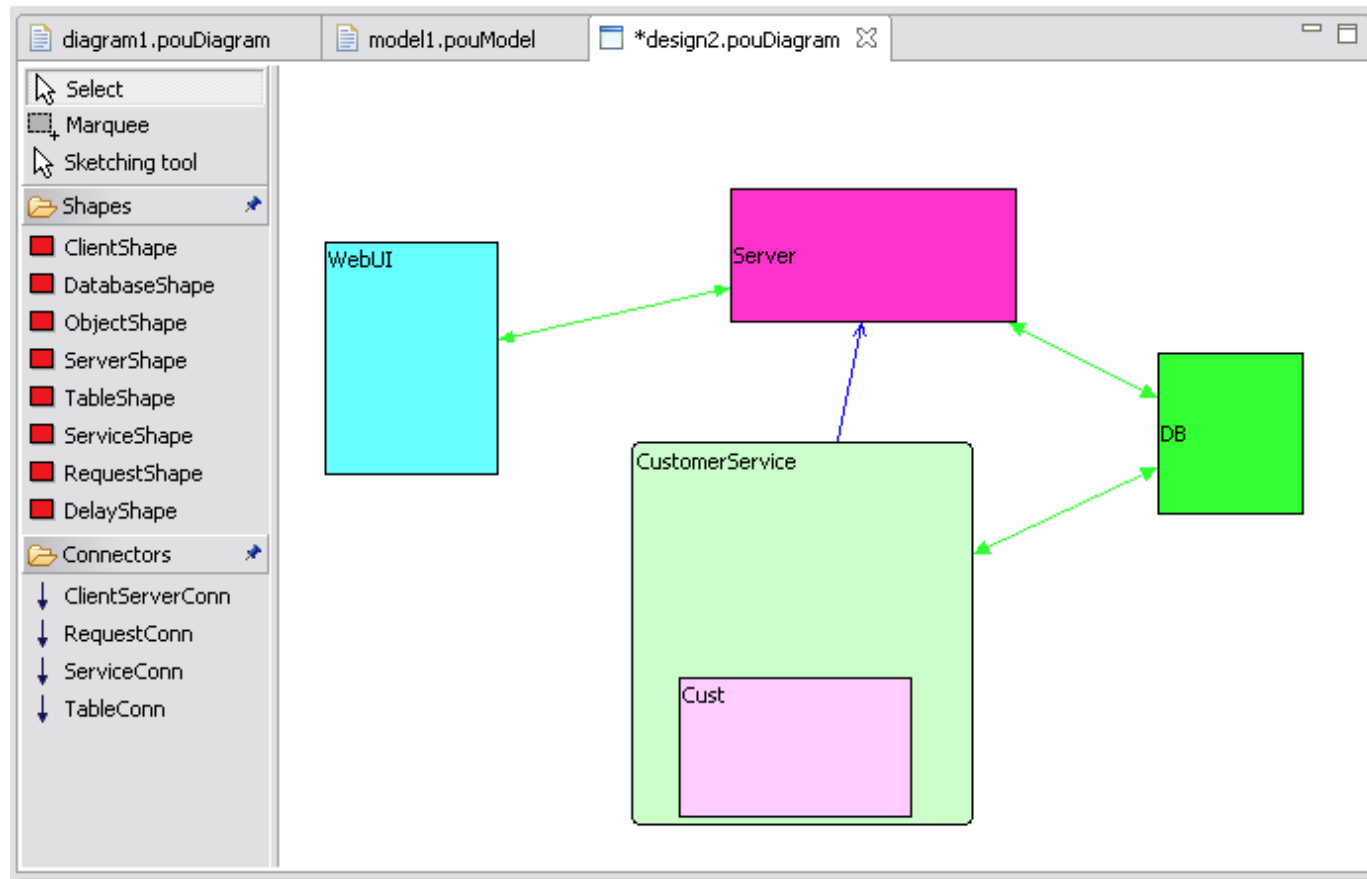
- ✓ For exploratory design/review?
 - Use pen & paper; Use whiteboard
 - Sketch architectural abstractions
 - Informally annotate
 - Discuss, revise via cross-out/rub-out/annotation
- ✓ This is a “natural, human-centred” interaction style
- ✓ But - what's kind-of difficult:
 - Capture (photocopy board? Digital smartboard?)
 - Distribute - in captured form (hard copy, scanned)
 - Modify - hard!!
 - Collaborate - same time/place only realistic...



Example #2



...then formalise in MTE tool



***BUT** ...very time-consuming, error-prone & now can't change with pen...!!*



ICSE 2007 - Sketching-based input & meta-tools

What we'd like to do in our design tools

- ✓ Combine “best of both worlds”
- ✓ Sketching-based input into MaramaMTE to support early-phase exploratory architecture design, design review
- ✓ Formalise “ink” into modelling components & text
- ✓ Computer-based management of ink and formalised diagram components
 - Can manipulate sketched ink with pen/mouse; save/load, distribute to other users etc
- ✓ Want to KEEP sketched content AND formalised content, keep consistent, and even sometimes show/manipulate together!
- ✓ Use sketching during e.g. reviews to augment collaboration support

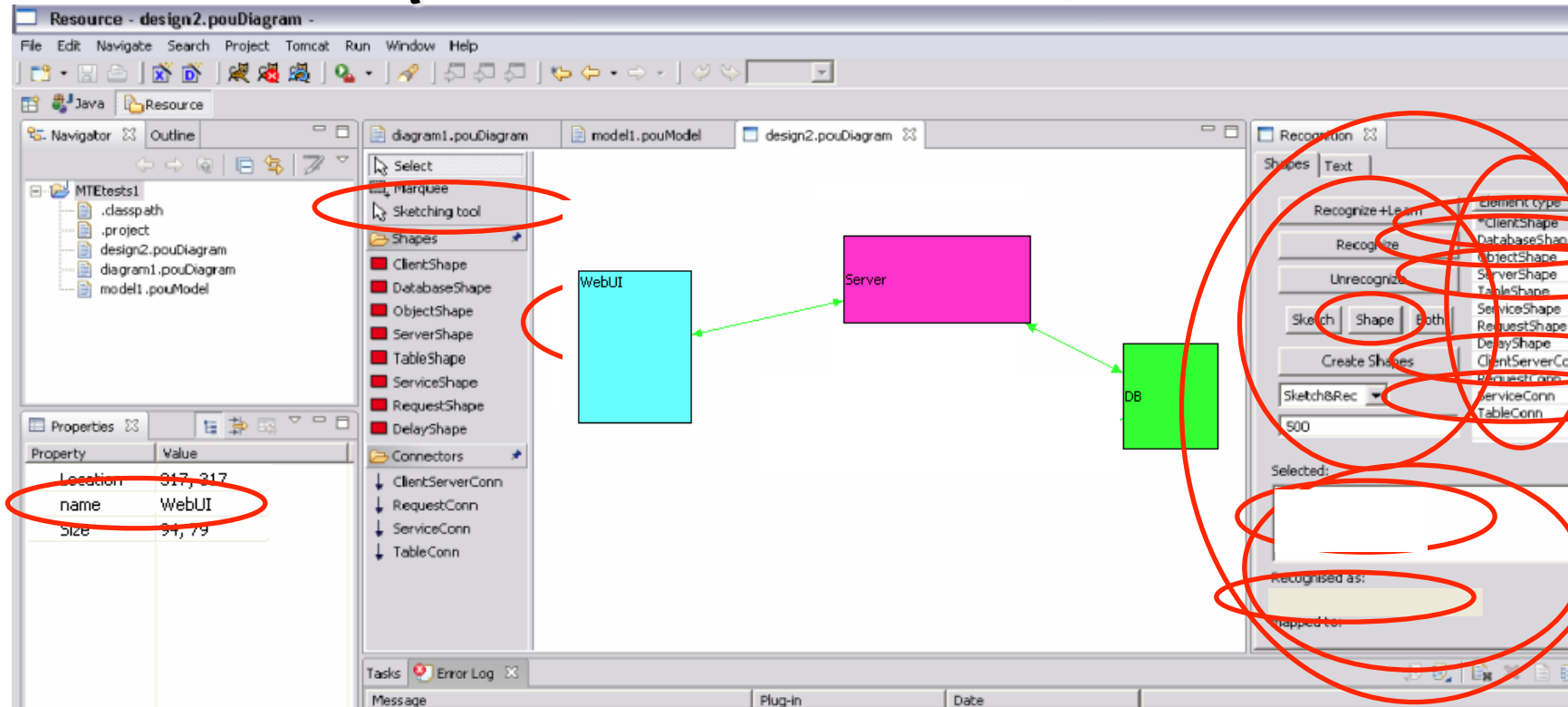


Adding sketching support to Marama-based design tools

- ✓ Marama - Eclipse-based meta-tools for domain-specific visual language tools (<http://www.cs.auckland.ac.nz/Nikau/marama/file/downloads.html>)
- ✓ Prototyped plug-in extension to Marama to add sketching support to any generated tool - NO CODE CHANGES to meta-tool ☺
 - "MaramaSketch" - works for ANY Marama-specified tool
- ✓ Augments Eclipse graphical editing with overlay & recogniser view
- ✓ Uses third-party stroke recogniser
- ✓ Use sketches to (1) formalise into Marama diagram elements; (2) support secondary annotation on diagram; (3) support collaborative review work
- ✓ Explored variety of interaction & recognition issues - so this work about both user interaction & software tool engineering issues...



Example #3 - Sketch & formalise

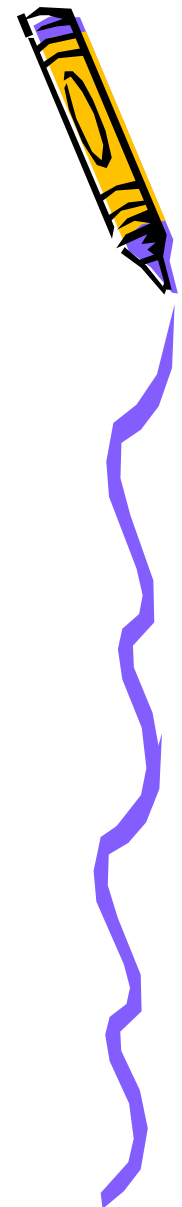
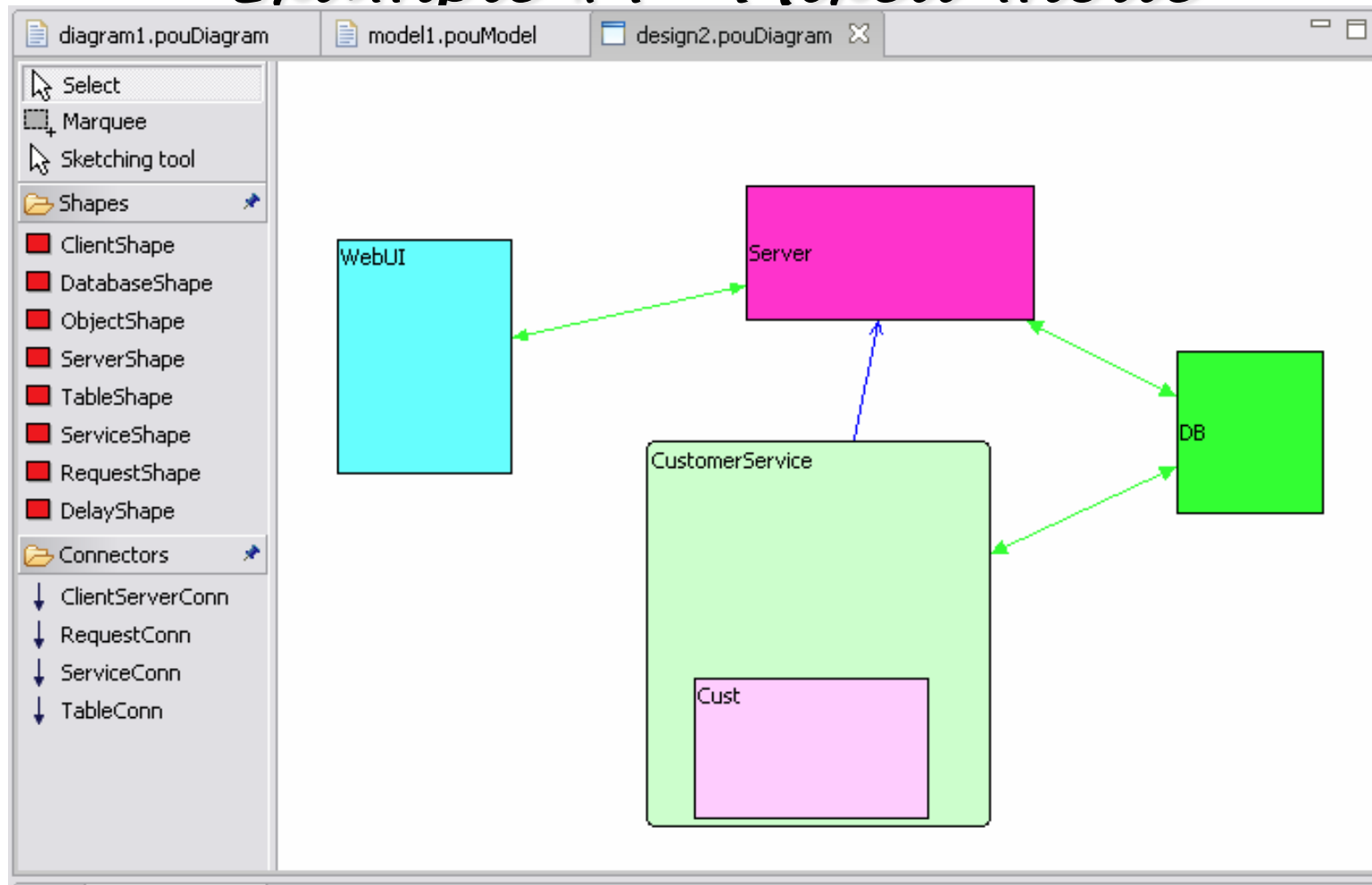


How cool is that...?! 😊

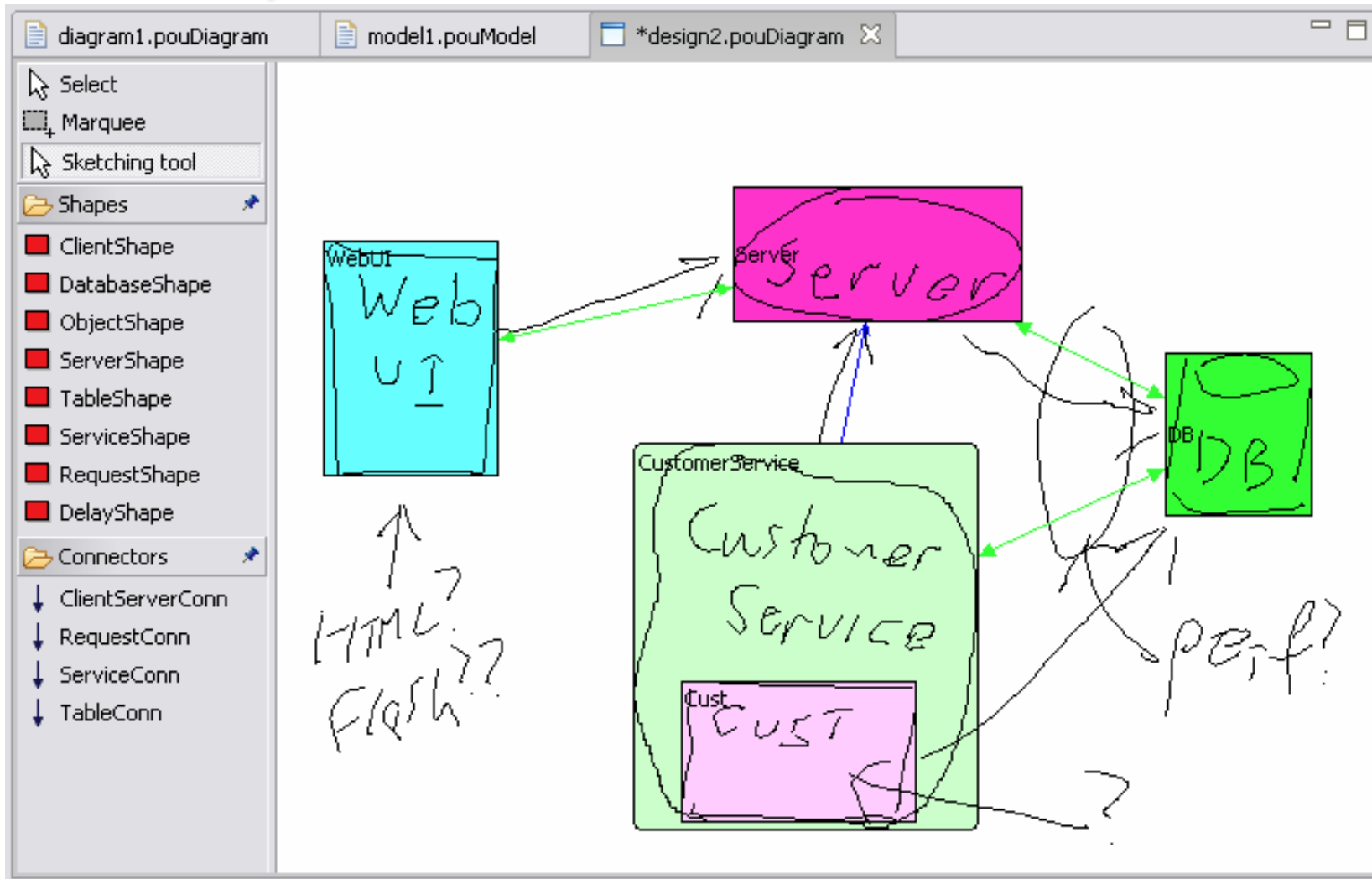


ICSE 2007 - Sketching-based
input & meta-tools

Example #4 - Mixed-mode



Example #5 - Review/Collaboration



Example #6 - Diffing/Other view types



The screenshot shows a software interface with a central diagram and several panels. A red circle highlights a list of actions on the right side of the diagram, and another red circle highlights a table in the Recognition panel.

Recognition Panel Table:

Element type	Prob
Action	46.0
Page	46.0
*ClientShape	844.0
Transition	125.0

Tasks Panel Table:

item	command	apply	undo
1	Move/resize Action Rectangle(320, 103, 62, 33)	true	true
2	Set property Transition -> Page.id = null	true	true
3	Set property Transition -> Page.probability = null	true	true
4	Set property Transition -> Page.id = null	true	true
5	Set property Transition -> Page.probability = null	true	true
6	Set property Transition -> Action.id = null	true	true
7	Set property Transition -> Action.probability = null	true	true
8	Move/resize ClientShape Rectangle(32, 40, 93, 67)	true	true



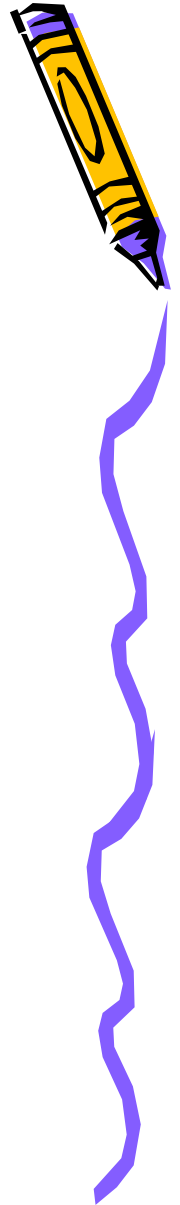
*A number of interesting issues arise
when adding sketching support...*

- ✓ *When do you recognise shapes?*
- ✓ *How do you recognise text vs shapes?*
- ✓ *Recognition - train vs generic?*
- ✓ *If train, WHEN do training?*



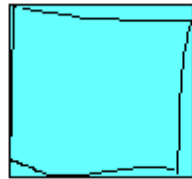
Issue #1 - when to recognise?

- ✓ *When do you recognise? Formalise?*
 - *Immediately draw stroke?*
 - *After set of strokes?*
 - *Recognise but leave sketch until all drawn?*
 - *Recognise all at once & formalise?*



Recognition Examples

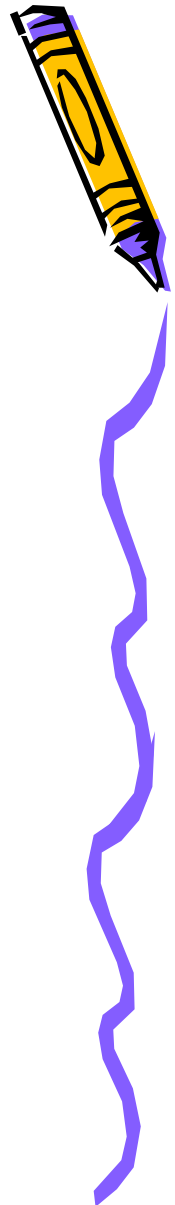
1. Sketch
2. Sketch & recognize/group
3. Sketch & create Marama shape



Selected:	
Recognised as:	
Mapped to:	

Selected:	1155605669921.6
Recognised as:	ClientShape
Mapped to:	

Selected:	1155605669921.12
Recognised as:	ClientShape
Mapped to:	ClientShape(1155605669921.13)

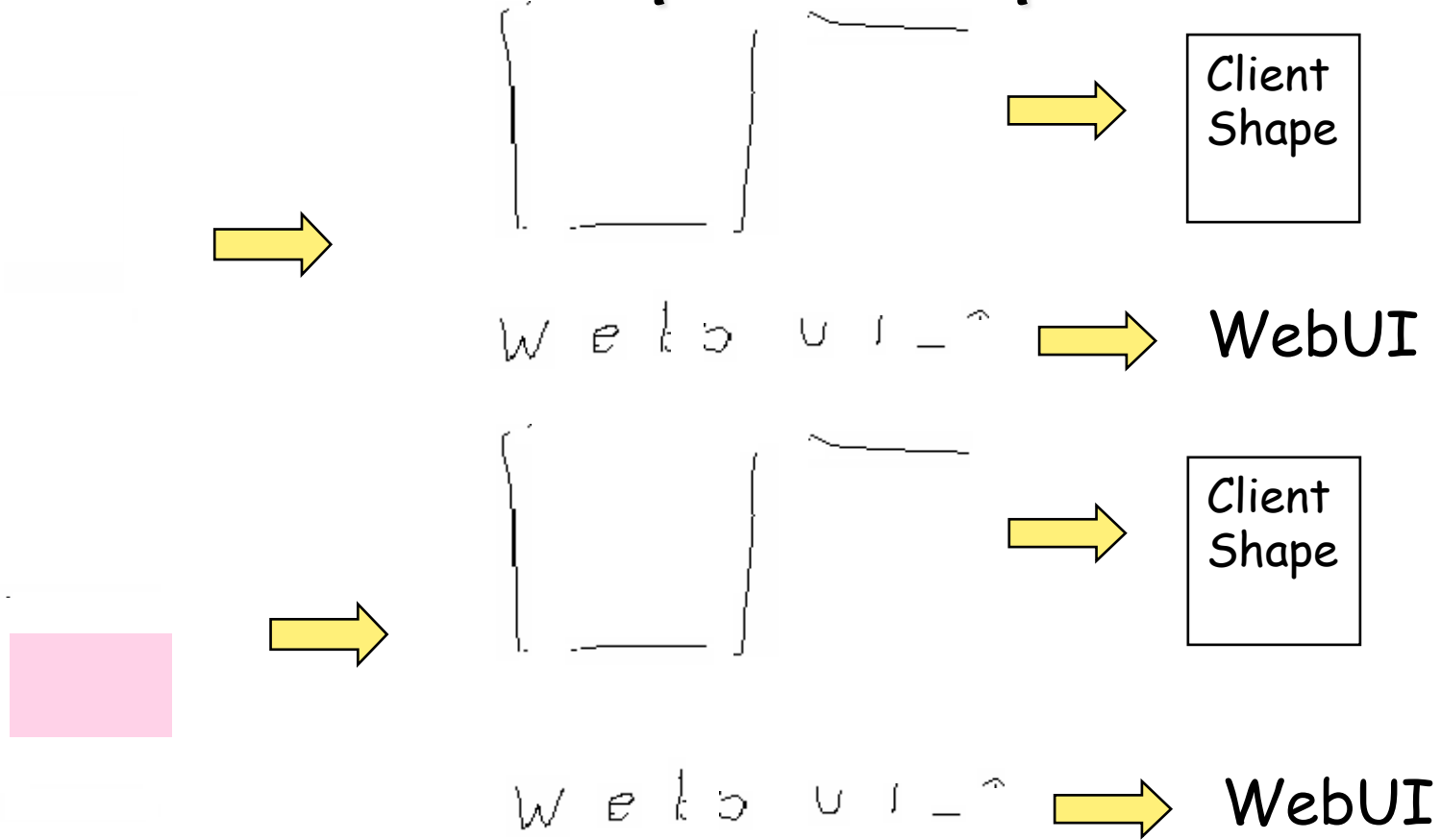


Issue #2 - text vs shapes

- ✓ Text & shapes suit different recognition algorithms
- ✓ Grouping of text vs shapes different e.g. linear/proximity/order (text) vs sequence/proximity/overlap (shapes)
- ✓ Options:
 - Attempt to classify strokes as text vs shape e.g. Plimmer's Inkit
 - Attempt to group strokes into single character; multiple characters vs single simple shape; multiple simple shapes
 - Have "area" where draw e.g. text, numbers e.g. Palm
 - Have pop-up drawing area for text e.g. PocketPC
 - Have annotations where draw text (our approach - based on our previous SUMLOW UML sketch tool work)



Text vs shapes examples

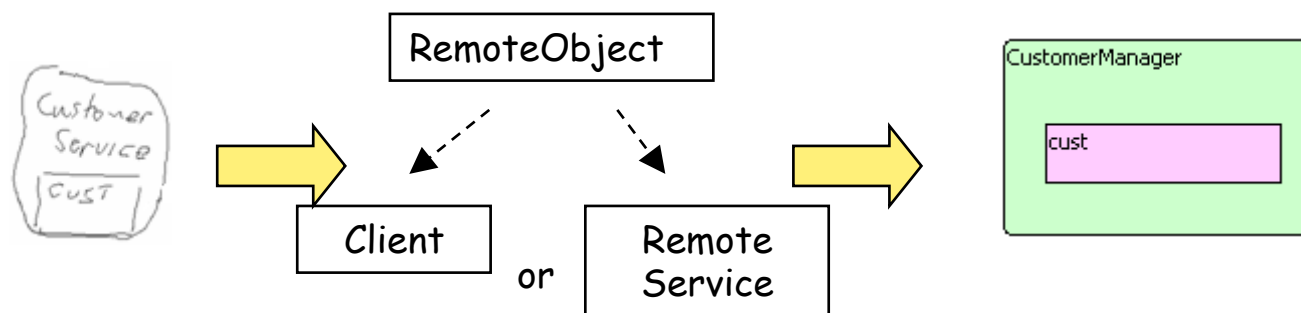
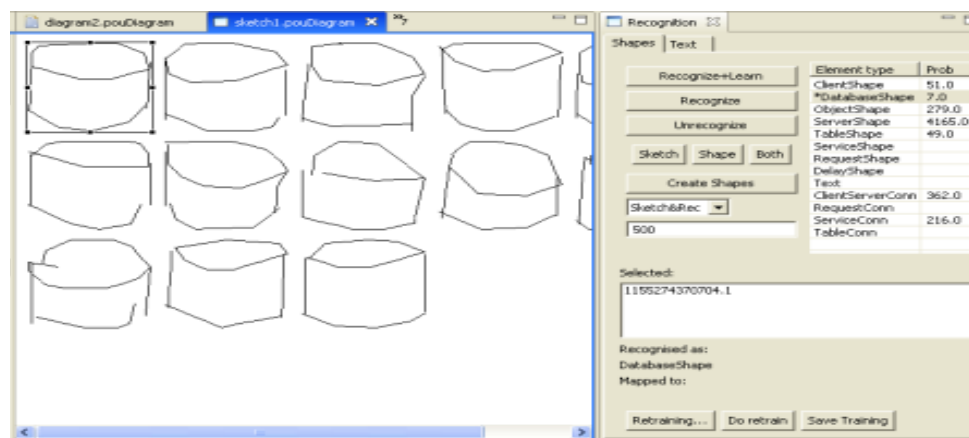
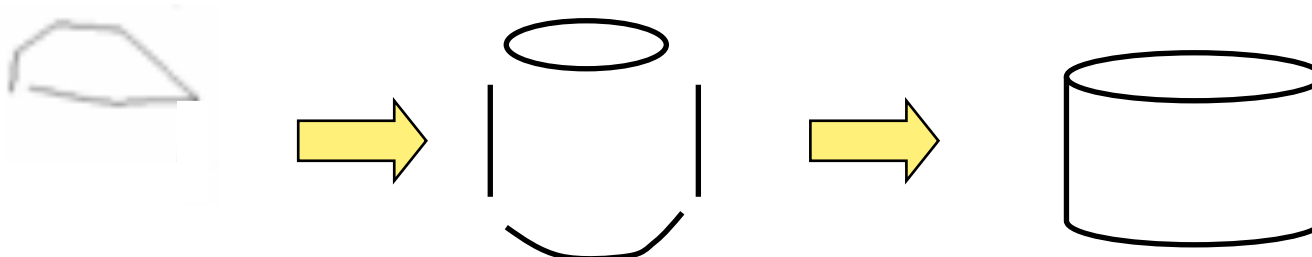


Issue #3 - use of training, context

- ✓ Can attempt to recognise shapes & text using “generic” properties e.g. Apert’s algorithm (geometric), TabletPC text
- ✓ Can use training-based e.g. Rubines algorithm (various ink properties)
- ✓ Can attempt to use “context” to infer text vs shape; shape classification e.g. if have syntax, use enclosure, proximity, etc



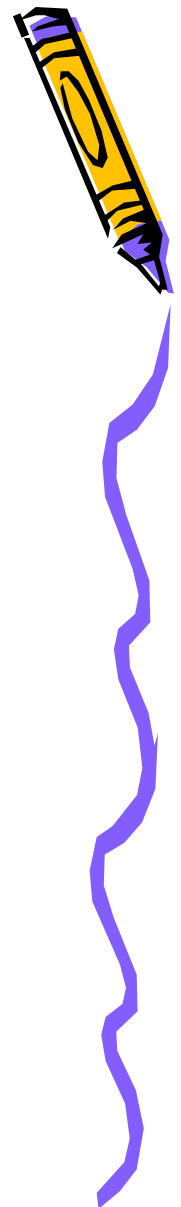
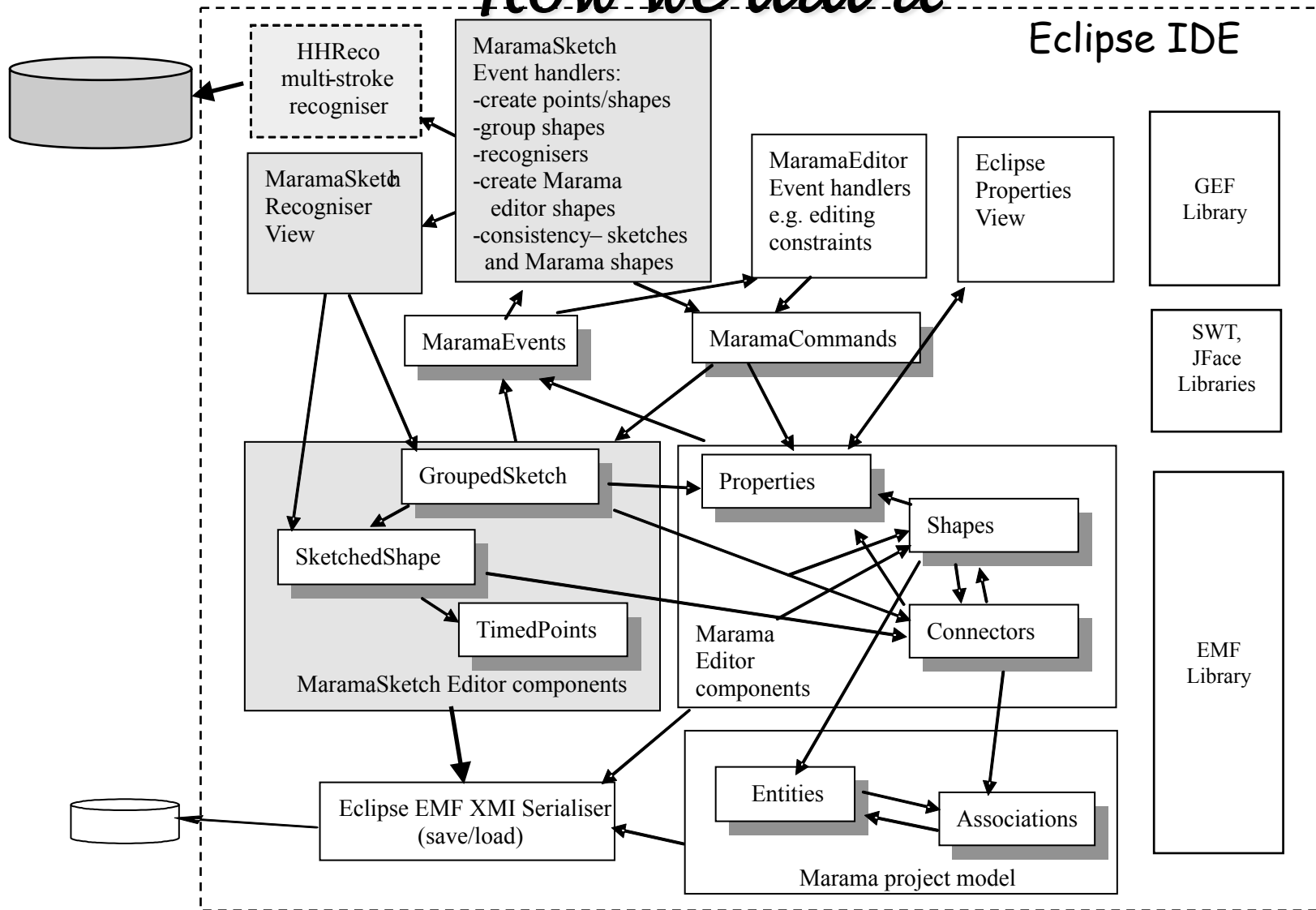
Geometrics vs training vs context



ICSE 2007 - Sketching-based input & meta-tools

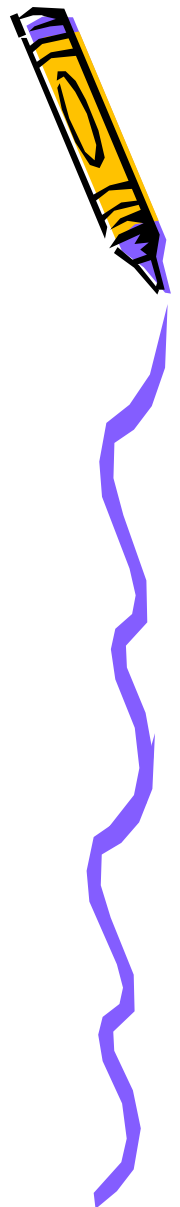


How we did it



Evaluation of prototype

- ✓ Very much exploratory prototype - rather clumsy interaction, complex Recogniser view, variable recognition reliability...
- ✓ Demoed to experienced users of Marama tools
 - Very positive reception
 - Like idea of "free" sketch support for any tool
 - Issues around "is it too flexible"??
- ✓ Cognitive Dimensions analysis
 - Analysis of key features & trade-offs
- ✓ (After fix divider/text recognition problems ☺) Conduct usability survey



Future work

- ✓ Add better shape/text division algorithm/ approach - Rachel Patel PhD...
- ✓ Achieve shape aggregation using meta-tool shape specifications
- ✓ Explore use of new Microsoft Tablet PC-based recogniser as plug-in component
- ✓ Explore use of better adaptive recognition algorithms



Conclusions

- ✓ Sketch-based input offers alternative interaction style
- ✓ Becoming more popular with e.g. E-whiteboards, Tablet PC, PDAs with stylus - WE NEED TO EXPLOIT THESE IN OUR SOFTWARE APPLICATIONS & TOOLS
- ✓ Wanted to explore ways to add to domain-specific visual language meta-tool and issues that arise
- ✓ Wanted test-bed for exploring different interaction, recognition, usage scenarios
- ✓ Added sketching layer via plug-ins to Marama meta-tools in Eclipse
- ✓ Many areas for future improvement to achieve goal of paper-like ease-of-use & software-like flexibility



References

- ✓ Grundy, J.C., Hosking, J.G., Li, N., Li, L., Ali, N.M., Huh, J. *Generating Domain-Specific Visual Language Tools from Abstract Visual Specifications*, *IEEE Transactions on Software Engineering*, vol. 39, no. 4, April 2013, pp. 487 - 515.
- ✓ Ali, N.M., Hosking, J.G., Grundy, J.C., *A Taxonomy and Mapping of Computer-based Critiquing Tools*, *IEEE Transactions on Software Engineering*, vol. 39, no. 11, November 2013, pp. 1494-1520.
- ✓ Zhu, N., Grundy, J.C., Hosking, J.G., Liu, N., Cao, S. and Mehra, A. *Pounamu: a meta-tool for exploratory domain-specific visual language tool development*, *Journal of Systems and Software*, Elsevier, vol. 80, no. 8, pp 1390-1407.
- ✓ 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.
- ✓ Cai, Y., Grundy, J.C. and Hosking, J.G. *Synthesizing Client Load Models for Performance Engineering via Web Crawling*, In *Proceedings of the 2007 IEEE/ACM International Conference on Automated Software Engineering*, Atlanta, Nov 5-9 2007, IEEE CS Press.
- ✓ Grundy, J.C., Cai, Y. and Liu, A. *SoftArch/MTE: Generating Distributed System Test-beds from High-level Software Architecture Descriptions*, *Automated Software Engineering*, Kluwer Academic Publishers, vol. 12, no. 1, January 2005, pp. 5-39.

