Kevlar
A User-Friendly, Type-Safe, Graphical Shell

A third year group project by
Tristan Allwood, Daniel Burke, Marc Hull, Ekaterina Itskova and Steve Zymler
Overview

• Background
• Motivation
• Specific usability issues solved
• Usability study
• Future work
• The next step
• Conclusions
Background

• What is a command shell?
• What is a pipeline?
• What is Kevlar?
Motivation

• What Kevlar solves:
  – Program discovery
  – Consistent Help
  – Argument validation
  – Typed pipes
  – Flexible piping
Hasn’t this been done before?

• PURSUIT
• VUFC
• Piper
• VisiQuest
• MEProf
Case Study: Image Manipulation

• Aim:
  – Load a directory of images
  – Resize the images to thumbnail size
  – Display the images before and after
The command line

ls *.png | sed ‘s\.*\.png$//’ | xargs --ri convert ‘{}’\.png --resize 100x100 /tmp/’{}’-thumb.png && gimp /tmp/*-thumb.png
Program Discovery

• What is the problem?
  – Finding the right tool for the job
  – Remembering the tool

• How does Kevlar solve this?
  – Sophisticated keyword-based search
  – Standardised help for programs
  – Categorised programs
  – Complete auto-complete
Argument Validation

• What is the problem?
  – Invalid arguments are not detected until execution-time

• How does Kevlar solve this?
  – Construction time parsing and validation of user arguments
  – Immediate visual feedback for mistakes
  – Execution disabled until the pipeline is valid
Typed Pipes

• What is the problem?
  – The command line only knows about binary data
  – There is nothing to stop users from creating a semantically invalid pipeline

• How does Kevlar solve this?
  – All pipes are typed and are valid for type safety
Flexible Piping

• What is the problem?
  – Programs can only have one or two useful input and output pipes

• How does Kevlar solve this?
  – Programs can have multiple named (and typed) input and output pipes
Kevlar: More than a shell?

- Macros
- Saving
- Loading
- History
- Dynamic addition of programs
Usability Study

• We set six tasks for users to complete
• We received feedback on the user experience
  – Some adjustments to the interface were made as direct feedback to this study
  – Ideas for future extensions were borne out of the study
Usability Study: Implemented Improvements

• “No feedback that the pipeline was saved”

• “if you RIGHT-click on the program, help should appear”

• “do NOT make us click on the circle to select it, clicking anywhere on the name should suffice”
  – With respect to program IO nodes.
Usability Study: Future Suggestions

- “Make it run fast =)”
  - Execution
  - GUI response

- “the way to connect different parts isn’t so clear.” (With respect to program IO node types)
  - Visualisation of types and type tree
  - Help finding programs based on IO types
Usability Study: Positive Feedback

• “I would use Kevlar for large tasks instead of a normal command line”
• “the program is very easy to use”
• “for tasks which take longer to execute the performance is ... acceptable, I felt”
• “great improvement over the classic command line”
Usability Study: Evaluation

• We do recognise our results come from a limited set of results, and most replies were from ‘console experts’
• This does not devalue the study as it has identified usability issues with Kevlar that we have had a chance to improve
Extension Work

• Plug-in-able visualisers
• Scripts and control structures
  – Loops, conditionals
• Arguments visible in Macros
• More Help
  – Tutorials, tool-tips, program discovery based on IO type
The Future of Kevlar

• Generic work-flow framework

• Distributed program discovery and execution

• Kevlar integration into Operating Systems?
Group Conclusions

- Communication a key factor to success
  - Negotiating interfaces
  - Group motivation
  - Teaming people together
- Importance of build systems and working environment
  - Constant integration
  - Instant feedback from other group members
Acknowledgements

- Paul Kelly
- Susan Eisenbach
- Robert Chatley
- Our usability study volunteers
- Matthew Sackman
- Eric
  - For being there