

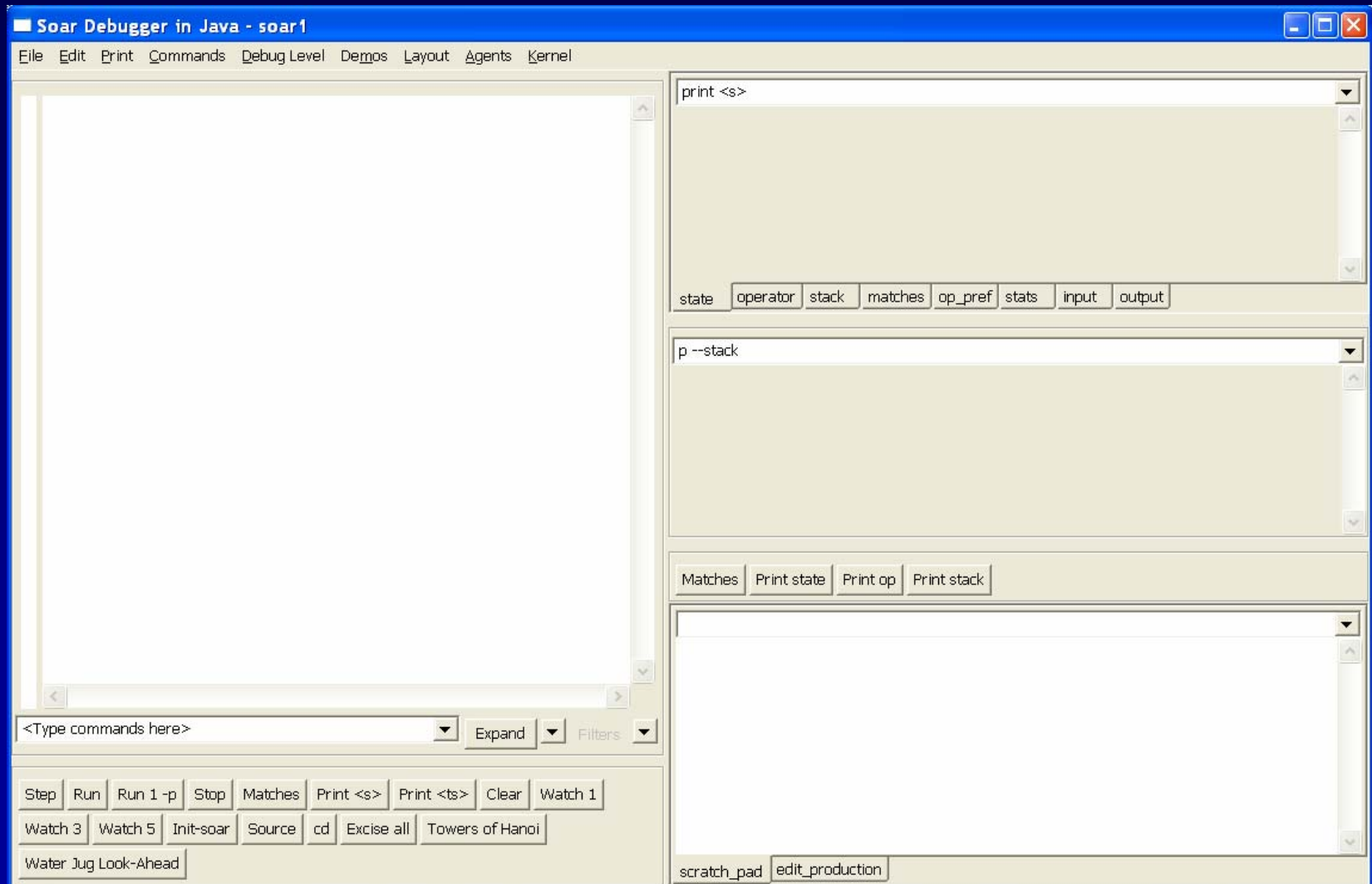
# The Inside Story on the Java Debugger

Douglas Pearson

[douglas.pearson@threepenny.net](mailto:douglas.pearson@threepenny.net)



# Reminder



# Design Principles

- XML based
  - Interface into Soar via XML (SML: Soar Markup Language)
  - But maintain high performance
- Plug-in Architecture
  - Debugger made up of a series of modules
  - Future extensions and user additions
- User configurable
  - Select elements you find useful
  - Combine as you like
  - Easier decision making on whether to include features

# Plug-in Architecture

- Simple plug-in approach
  - Each window is instance of a separate class
  - All derive from abstract base class
  - Rest of code deals only with the base class
- SWT based
  - May help with Eclipse integration
  - Eclipse defines a full OSGI plug-in model etc.
- Building a plug-in module is pretty simple
  - Hope user community will build/modify them

# Existing modules

- **AbstractView (base)**
  - **AbstractComboView** (combo box for commands)
    - **FoldingTextView** (tree trace)
    - **TextTraceView** (text trace)
    - **UpdateCommandView** (updating window)
  - **ButtonView** (button panel)
  - **EditorView** (edit production)
- **Extend from AbstractView or an existing class**
  - No need to be text based

# Each View's Responsibilities

- Register for events it is interested in
  - E.g. Listen for trace output events
  - Multiple windows listening for same event is efficient
- Store and retrieve its internal settings (as XML)
  - E.g. command history; button names; specific properties
  - Stored as part of a layout file (.dlf)
- Implement its windows within an SWT Composite parent window
  - Can do anything you like inside that
- Optional:
  - Support execution and display of commands (e.g. from menus/buttons)
  - Support entry of commands or other parameters from user (e.g. combo box)
  - Support find
  - Support user configurable properties

# AbstractView structure

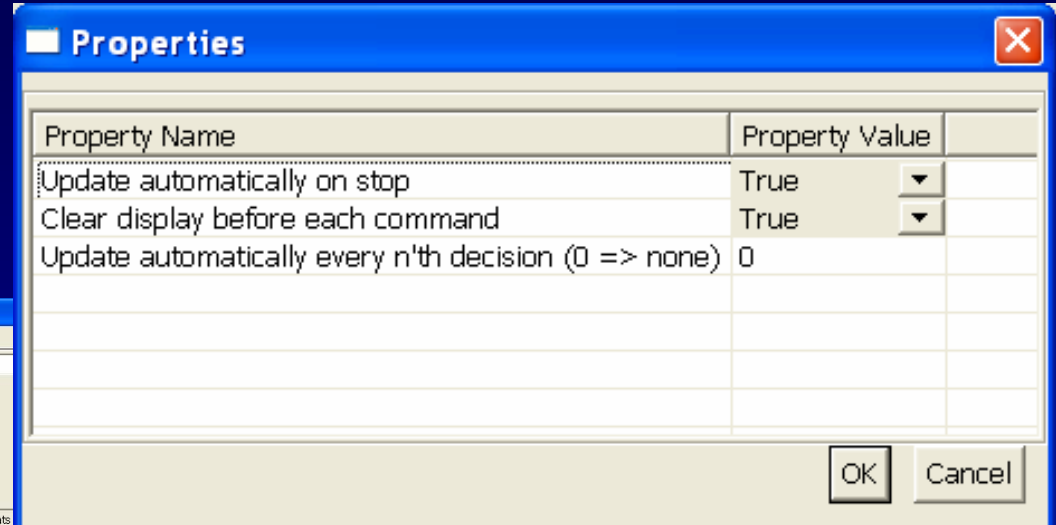
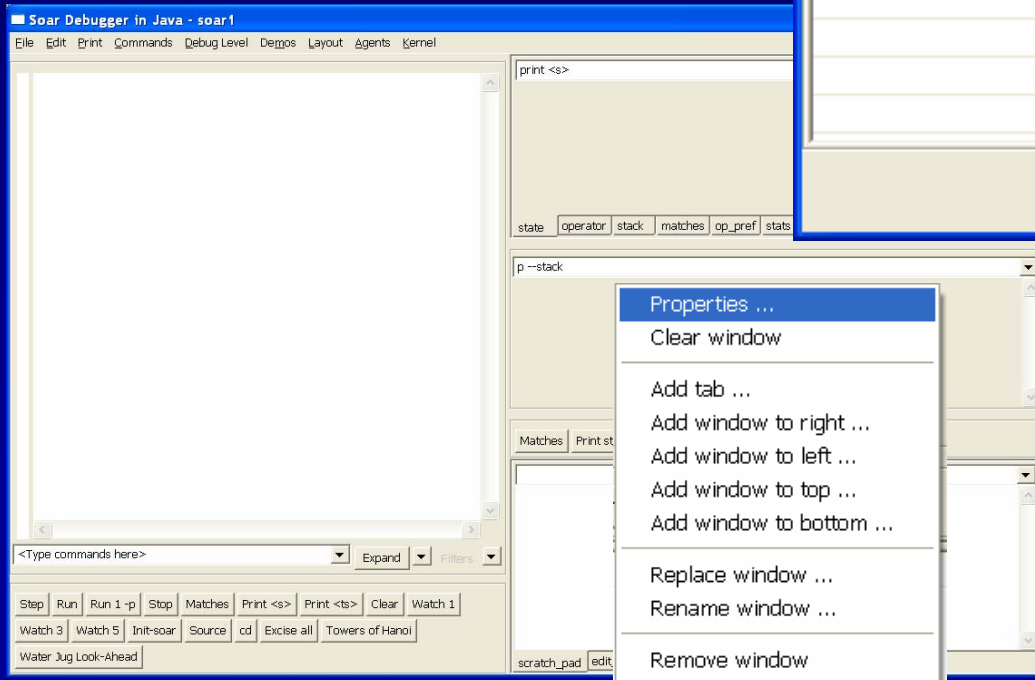
- Event handling
  - void registerForAgentEvents(Agent agent) ;
  - void unregisterForAgentEvents(Agent agent) ;
- Command execution
  - String executeAgentCommand(String command, boolean echoCommand) ;
  - void displayText(String text) ;
- Initialization and storage
  - void init(MainFrame frame, Document doc, Pane parentPane) ;
  - ElementXML convertToXML(String tagName, boolean storeContent) ;
  - void loadFromXML(MainFrame frame, Document doc, Pane parent, ElementXML element)
- Actions
  - void showProperties() ;
  - boolean find(String text, boolean searchDown, boolean matchCase, boolean wrap, boolean searchHiddenText) ;
  - void fillInContextMenu(Menu contextMenu, Control control, int mouseX, int mouseY) ;

# User configurable

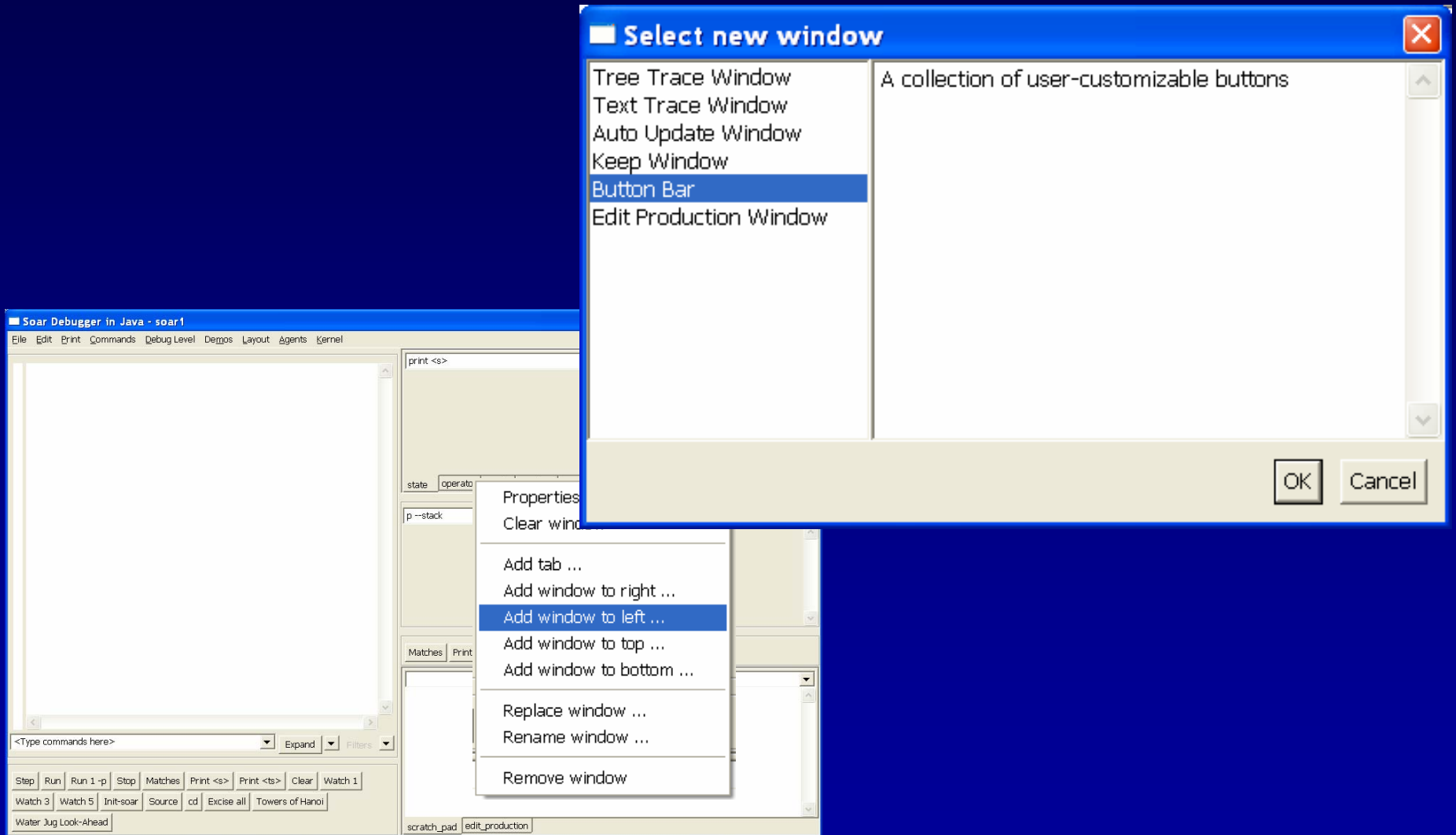
- Changing the display w/o recompiling
  - Add/remove windows (plug-in modules)
  - Adjust properties to change behavior
    - E.g. update window every n-th decision
- Can save multiple window layouts
  - Please send us layouts you like, we'll share them



# Configuring Window Properties



# Adding/Removing Windows



# Adding/Removing Buttons

The screenshot illustrates the process of configuring a button in a software application. The main window shows a command line with "p --stack" and a toolbar with buttons: Help, Init-soar, Step, Run 1 -e, Run, Stop, and Start up. A context menu is open over the "Start up" button, with "Edit button..." selected. The "Properties" dialog is open, showing a table of properties:

Property Name	Property Value
Label	Start up
Command to execute	watch 3...
Internal command to execute (advanced)	

The "Enter button actions" dialog is also open, showing a list of actions:

```
Actions watch 3
watch --learning print
source "e:\soarmich\test.soar"
```

Buttons for "OK" and "Cancel" are visible in both dialog boxes.

# Nuggets and Coal

- Nuggets
  - Java
  - Plug-in approach works
  - Customization works
- Coal
  - SWT can be tricky but the results look good
  - Plug-in framework needs to mature
  - Interaction between plug-ins will need extension
  - Not integrated with Eclipse yet
- Want to try a plug-in? Please let us know:  
[soar-sml-list@umich.edu](mailto:soar-sml-list@umich.edu)