Why “Common Sense” Isn’t Enough

- Performance tools can be hard to implement
  ... but it’s tool usability that’s the hardest
- Tools must reduce execution data to manageable size
  - Filter out redundant or unnecessary data
  - Extrapolate higher-level events/objects from low-level data
- Tools must minimize
  - The possibility for user error
  - The amount of effort to find information
- Tools must present info in ways that are meaningful to users
This Presentation

- Examples drawn from parallel and grid computing
- Also apply to
  - Serial computing (if considering “suites” of executions)
  - Cluster computing
  - Distributed computing
- Current techniques that support usability
  - Visual representations
  - Direct manipulation
- What’s needed to make future tools really usable

Graphical Techniques Provide a Key

- Non-textual techniques
  - Rely on graphical attributes – shape, color, texture
  - Figurative or symbolic representation of objects, characteristics
- Visual displays of quantitative data
  - Can make large data domains coherent
  - Can encourage the eye to compare/contrast elements
  - Can expose anomalies
  - Can reveal information at varying levels of detail
  - Can reflect both statistical and logical nature of data
What Visualization Offers

(1) Provides way to manage voluminous / complex data
(2) Capitalizes on user’s pattern recognition capabilities
(3) Helps user “interpret” program behavior

Adding Direct Manipulation

- Ability to grab/click objects to change the nature of the display

<table>
<thead>
<tr>
<th>Logical Directness</th>
<th>Physical Directness</th>
</tr>
</thead>
<tbody>
<tr>
<td>least direct</td>
<td>most direct</td>
</tr>
<tr>
<td>cascaded menu</td>
<td>pointer device</td>
</tr>
<tr>
<td>menu</td>
<td>touch screen</td>
</tr>
<tr>
<td>button or icon</td>
<td>haptic (virtual reality)</td>
</tr>
<tr>
<td>select a graphical object</td>
<td>glove</td>
</tr>
<tr>
<td>transform a graphical object</td>
<td></td>
</tr>
</tbody>
</table>

Northwest Alliance for Computational Science & Engineering
What Direct Manipulation Offers

(1) Provides way to manage interrelated displays or operations
(2) Establishes clear “connections” between tool elements
(3) Gives user a sense of control
(4) Reduces opportunities for errors due to misinterpretation

What Current Tools Can Do

- Tools use visualization and direct manipulation to:
  - Highlight key information to help user focus attention
  - Reveal information about different aspects of performance
  - Help navigation through large codes
  - Help navigation through large performance data
Help User Focus Attention

CXperf (HP)
Sort (or filter) so that most important data is most obvious
Eliminate extraneous info, making it available through options / menus / popups

Help User Focus Attention

Tau (Univ. of Oregon)
List “top resource users” to help user concentrate improvement efforts
Support Multiple Perspectives

**jumpshot (Lusk)**
- Provide obvious ways to view different data subsets
- Support multiple, orthogonal sets of controls

**CXperf (HP)**
- Show slices through data space, rather than everything
- Provide simple, obvious controls for cutting other slices
- Allow user to re-sort or filter what's shown
Support Navigation through Large Codes

Xprofiler (IBM)
Provide “overview” display, giving sense of context
Allow user to collapse or hide portions of the display, in successive layers

Xprofiler's thumb-nail images as navigational aids:
- Give user sense of context
- Support fast movement
Support Navigation through Large Codes

**DEEP (Pacific Sierra)**
- Graphical synopses offer quick recognition or summary.
- Can be basis for filtering or selection.

Support Navigation of Large Data Space

**Jumpshot (Argonne NL)**
- "Preview" lets user navigate quickly through time.
Current Tools Aren’t Enough

- Users need to approach performance data flexibly
  - Focus different “perspectives” independently
  - Experiment with different combinations of performance factors
  - Pose basic questions about performance
  - Play “what-if” to estimate effectiveness of performance improvements

- Current tools portray information, but aren’t flexible enough to do what users really want

What Users Really Want …
What Performance Tools Need to Achieve

The public utilities analogy …

1. The resources consumers want – when they want
2. Consumers don’t care how it really works
3. Consumers pay for what they use

<table>
<thead>
<tr>
<th>Electrical Power</th>
<th>Telephones</th>
<th>Parallel/Cluster /Grid Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 or 2 choices</td>
<td>10-20 options</td>
<td>100s-millions</td>
</tr>
<tr>
<td>Plug it in</td>
<td>Dial 7 or 10 or 11 or ~15 or (N+14) digits</td>
<td>Few clicks to total rewrite</td>
</tr>
<tr>
<td>Charged for usage</td>
<td>Charged fee + (fee + usage) + (fee+usage+premium)</td>
<td>Open issue (but will be complex)</td>
</tr>
</tbody>
</table>

New Requirements for Performance Tools

- Users need to approach performance data flexibly
  - Multiple perspectives that can be focused independently
  - Easy to experiment with different combinations
  - Easy to maintain a sense of context and control

New Requirement #1:

Extend navigation for size/complexity of today's performance environments
Navigation Through Very Large Data

- **Thumbnail** and **strip mural images** are critical
  - Provide context
  - Support movement

[Jerding & Stasko, Ga Tech]

Navigation Through Very Large Data (2)

- **Textual labels must adapt to user’s “focus of attention”**
  - As cursor moves, labels appear disappear
  - Avoids clutter

Suggested enhancements of AIMS (Yan et al.)
Navigation Through Complex Environments

- Tools must reflect growing complexity of performance environments
  - Machine heterogeneity
  - Bandwidth requirements of data
  - Variation in resource costs

[Abramson/Buyya/Giddy, Monash U]

Navigation Through Complex Source Code

- Tree maps clarify nesting hierarchies
  - Can be clearer than graph representations
  - Combine well with thumbnail and/or zooming techniques
Navigation Through Complex Source Code

- Structure diagrams portray instances of source objects
  - Clearer than graphs
  - Combine well with thumbnails

Jinsight (IBM Research)

Navigation Through Multidimensional Data

- Portray distinct levels of detail (not just magnifications)

  - Zooming in provides more detail over smaller data space
  - Zooming out reduces detail
Navigation in Multidimensional Data (2)

SPV
(Intel)

Integrating Multidimensional Data

- N-dimensional data require the ability to move through multiple data spaces

- 3-D graph viewers
  - Can support movement across spaces
  - Could be combined w/ individual strip murals

[Vinus, Koike & Yoshihara]
New Requirements for Performance Tools

- Users need more help in performing their most common tasks
  - Comparing data across multiple executions
  - Comparing across multiple dimensions of data

New Requirement #2:
Tools should reflect key user tasks, not just the data available

Comparison Is the #1 User Task

- Currently requires that user
  - Invoke multiple instances of tool
  - Manually synchronize views

Java Instrumentation Suite + Paraver (Torres, CEPBA)
Comparing Within an Execution

- User repeatedly revisits areas of code or performance data
  - “User landmarks” highlight locations for later re-use

Comparing Across Executions

- Should be able to compare executions automatically
  - Visualize
    - Value changes
    - Performance changes
Comparing Across Data Dimensions

- Synchronized summaries
  - Selection in one data space triggers corresponding movement in another

Can also use thumbnails/murals to integrate distinct types of data

[Lightweight Corefile Browser, Ptools]
Helping the User Focus Attention

- Highlighting can be more than a binary property
- **Weighting-based fisheye views**
  - Prominence of element scaled to its calculated “importance” or “cost”

[Koike & Yoshihara]

New Requirements for Performance Tools

- The burden is still on the user
  - To compare information and metrics
  - To estimate “goodness” of performance
  - To guess the likely effects of different program modifications

**New Requirement #3:**

Incorporate automatic analysis and prediction methods
Apply Statistical Analyses

- Lists of “top resource-users” alone aren’t much help
  - Cluster analysis
  - Sensitivity analysis
  - Regression analysis

Estimate Performance Quality

- Use colors to provide sense of “value”
  - Convey goodness or badness
  - Convey intensity (bright vs. dim color)

S-check (NIST)

P3T (Fahringer et al.)
Point the User to a Solution

- Merlin maps out what an “expert” would try if he/she had those results

Identify Patterns in Performance Data

- Paradyn recognizes some patterns of memory access
Export Data to Other Analysis Tools

- Ursa Minor exports its data to statistical tools, spreadsheets, databases, etc.

New Requirements for Performance Tools

- Visualization and direct manipulation have become aren’t as usable as they should be
  - Don’t make information about visual operations clear enough
  - Don’t organize visual operations well
  - Don’t distinguish visual operations from other operations

New Requirement #4:
Make visual techniques self-explanatory
Remind Users that Operations Are Possible

LCB (Par.Tools Cons)
Use message text to let user know what can be manipulated
Change the cursor when it can be used to manipulate objects

Organize Operations in Pairs

CXperf (HP)
If rubber-band+click causes area to “contract”…
click on contracted area should “expand” it
Use Distinct Mechanisms

If rubber-band + click control which routines are hidden…
Use menus, radio buttons, etc. to filter out threads or to choose which metric is shown

Phrase Options in Terms of User Tasks

- Current use terms describing tool mechanisms
Conclusions

- Today’s performance tools include a range of useful techniques
  - Visualization
  - Direct manipulation
- But they still focus too much on “showing data” rather than “helping users”
  - Navigation and synchronization aids
  - Direct support for comparison and analysis
  - Automatic analysis

What’s Really Wrong with Today’s Tools?

- Users say performance tools make it too hard to carry out common tasks

  "Tools are supposed to make me more productive, but I can usually find the problem quicker with PRINT and calls to timers."

  "I want to ask simple questions -- why do tools make them so hard to answer?"
Tools Must Match User Needs and Habits

A tool that doesn’t really help the user with key tasks…

… may be worse than no tool at all!

Mozart writing the digital version of his symphony No. 38 in D major