Using Visualization to Support Network and Application Management in a Data Center: Visual-I

Danyel Fisher
David A. Maltz
Albert Greenberg
Xiaoyu Wang
Heather Warncke
George Robertson
Mary Czerwinski

Microsoft Research
The Tier-1 Operator’s Challenge

• An analyst in the SOC:
  – Hundreds of properties,
  – With a thousand events a day,
  – Across hundreds of thousands of servers,
  – With a configuration that constantly in flux,
  – And analysts who are experts in small verticals at best…

• Incidents mean lost customers and lost revenue: time to repair critical
  – Decide who to involve
  – Decide what to do
The Tier-2 Operator’s Challenge

• An expert in operating a single property:
  – Deep experience with the dynamic behavior of a property
  – Not a developer – doesn’t know the code; empirical, black-box knowledge
  – Responsible for running, upgrade, restoration of service
Tier-2 Tools Today

Need to check many places to figure out what is happening
Analysis of Why Tools Fail

• **Single Perspective:** “tool per component” model taxes operators
  – One tool for a machine’s details, another tool for its connections, and query a database for its status.
  – Operators responsible for carrying context between tools (e.g., name of server)
  – Can’t see the forest for the trees (“the service for the servers”)

• **Abstraction failure:** Operators need abstractions for “chunking”
  – Operators think in hierarchies and topologies

• **Bad/Inconsistent data:** Tools must help operators cope with noise
  – Must accommodate occasional out of range or non-compliant data.
  – Threshold rules/alerts don’t work – give operator more information
  – Must cope with inconsistent meta-data

Concrete things:

• ** Scalability:** tree-views don’t work for more than 100 servers
• ** Monitoring overhead:** Every tool wants its own monitoring, but monitoring is already there
Visual-I Goals

Overcome the failings of previous tools

Philosophy:
• Use visualization to make instantly salient:
  – The structure of a property
  – The relationships among its parts and with other properties
• Leverage this structure to help user navigate data:
  – Request paths, volume
  – Who last worked on the box, maintenance and upgrade status
  – Temperature, CPU/Memory Utilization

Concrete things:
• Leverage existing data sources (SQL, csv, etc.)
• Be rapidly reconfigurable by the operations team itself
Data, Model View System

Model: front-end server cluster

Model: back-end server cluster

Model: connection

Visualizer for list-of-machines

MachineViz

MachineViz

MachineViz

Visualizer for list-of-machines

Visualizer for list-of-sets-of-machine-database-pairs

Server

DB

DB

DB

Model: front-end server cluster

Topography data source query and connection

Status data source query and connection for CPU

Status data source query and connection for status

Server

DB

DB

DB

Model: back-end server cluster

Model: back-end server cluster

Model: connection

Text files

Status database

Status database

Production servers

Monitoring and Aggregation Tool

Data, Model View System
<SqlDataSource Id="BackEndStatus"
    Interpolation="LastKnown"
    ObjectKey="{server}"
    TimestampKey="{time}"
    AttributeKey="{counter}"
    ValueKey="{value}">
    <ConnectionString>Server=msr-3d-demo14;Database=O….</ConnectionString>
    <Query>
        SELECT [server], [time], [counter], [value]
        FROM  ABCHData p
        WHERE  p.[server] like '%sql%' AND p.[counter] IN ('…')
    </Query>
</SqlDataSource>

<MappedModelBuilder Id="ClusterBuilder_CSV">
    <Mapping>
        <MachineCluster Id="BaseGroup" DisplayName="ABCH Back End">
            <MachineCluster Id="{AC}" DisplayName="{AC}"
                <MachineSubCluster Id="{ClusterName}" DisplayName="{ClusterName}"
                    <DatabaseServer Id="{ServerName}" DisplayName="{ServerName}" AC="{AC}">
                        </DatabaseServer>
                    </MachineSubCluster>
                </MachineCluster>
        </MachineCluster>
    </Mapping>
</MappedModelBuilder>
<MappingGroup Id="abch_map">
  <Mappings>
    <Mapping MapFrom="{CPU}" MapTo="ColorIndicator">
      Mapper="SteppedColorMapper"
      MapperArgs="Steps=[20,50,70,100];Colors=[Blue,Green,Orange,Red]"/>
    </Mapping>
    <Mapping MapFrom="{Trx/Sec}" MapTo="CenterCircleColor">
      Mapper="SteppedColorMapper"
      MapperArgs="Steps=[2000,3000,4000,10000];Colors=[Blue,Green,Orange,Red]"/>
  </Mappings>
</MappingGroup>

<ModelVisualizations>
  <ModelVisualization ModelId="abch_be_csv" VisualizationId="viz2">
    <StatusMappings>
      <Status MappingGroup="abch_map" DataSource="BackEndStatus">
        UpdateFrequencySeconds="10"
      </Status>
    </StatusMappings>
  </ModelVisualization>
</ModelVisualizations>
Visual-I Display
Visual-I History Map
Visual-I Display
Visual-I Display
<table>
<thead>
<tr>
<th>SQL478</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL479</td>
<td>1</td>
</tr>
<tr>
<td>SQL480</td>
<td>2</td>
</tr>
<tr>
<td>SQL481</td>
<td>3</td>
</tr>
<tr>
<td>SQL482</td>
<td>4</td>
</tr>
<tr>
<td>SQL483</td>
<td>5</td>
</tr>
<tr>
<td>SQL484</td>
<td>6</td>
</tr>
<tr>
<td>SQL485</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>31</td>
</tr>
</tbody>
</table>
Some Future Directions

• Coping with meta-data anomalies through multiple interpretations
• Better visualizations for relationships between components
  – Current visualizations get too busy if lines are used
  – Not clear if box layouts are general enough
• Longitudinal study of Visual-I in use
Summary

Visualization tools can help operators:

- **Understand** the current state of the system
  - Even when the system is in an inconsistent or an unusual state
- **Discover correlated behaviors** critical to debugging the system
- **Deal with inconsistencies**
  - Rules and autonomic tools have difficulty when the structure of the application does not fit the assumptions of the management system

Key features to provide are:

- Make the visuals follow the operator’s mental models of the system
- Enable correlation by providing context
  - Show same item but multiple instants in time
  - Show multiple items but functionally related