Enterprise Application Discovery and Monitoring Management with ExplorViz

Master’s Thesis Presentation

March 21, 2018

Alexander Krause – Kiel University
Motivation
Motivation

ADAMMS
(Application Discovery and Monitoring Management System)
Design – Application Discovery

Application Discovery (Single Iteration)

1. Obtain OSJPL
   - current OSJPL

2. Obtain Working Directory
   - enriched OSJPL (1)

3. Apply Recognition Strategies
   - enriched OSJPL (2)

4. Analyze and Merge
   - old OSJPL

OSJPL – Operating System Java Process List

<<Interface>>
RecognitionStrategy
- isDesiredApplication() : boolean
- applyStrategy() : void
Design – Monitoring Management

- Kill process
- Insert identifier
- Update model
- Insert Monitoring
- Find process
- Internal model
- Get new OSJPL
- Run exec cmd

java
- -javaagent:/kieker.jar -Dexplorviz.agent.model.id=2
- -cp -jar /kiekerSampleApp/sampleApplication.jar
Architecture – Component Design
Architecture – Component Design
Implementation – Software Stack

Agent
- Jersey
- Jersey Client

Backend Ext.
- Jersey
- Jersey Client

Frontend Ext.
- Cytoscape.js

HTTP (REST) 1..*

HTTP (REST) 1..*

HTTP (REST) 1..*
Implementation – Overview Page

Tomcat Web Server - General Information

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is stopped?</td>
<td>false</td>
</tr>
<tr>
<td>Working Directory</td>
<td>/</td>
</tr>
</tbody>
</table>

User Execution Command: Use-OS-Exeo-CMD

Agend Execution Command

OS (initial) Execution Command

Tomcat Web Server - Monitoring Information

[Buttons: Save, Restart, Stop]
Evaluation

Usability Evaluation

Setup

Results & Discussion

Goals
- Overall usability regarding setup and operation
- Realization of usability characteristics

Experiment
- Proband solved tasks
- Conductor notated issues

Structural Interview
- Perceived usability
- Enhancements
Evaluation

Usability Evaluation

Setup

Results & Discussion

<table>
<thead>
<tr>
<th>Name</th>
<th>Quantity</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>node2, node3</td>
<td>2</td>
<td>CPU: 2x Intel Xeon E5-2650 (2.8GHz, 8 cores)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ram: 128 GB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OS: Debian</td>
</tr>
<tr>
<td>lp1</td>
<td>1</td>
<td>CPU: Intel Core i7-6700HQ</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ram: 32 GB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OS: Windows 10</td>
</tr>
<tr>
<td>lp2</td>
<td>1</td>
<td>CPU: Intel Core i5-4278U</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ram: 16 GB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OS: macOS High Sierra</td>
</tr>
</tbody>
</table>

Software Landscape

- 9 running applications
  - Tomcat with JPetStore 6
  - kiekerSampleApplication

1 https://github.com/czirkelbach/kiekerSampleApplication
Evaluation

### Usability Evaluation

- **Learnability**
  - Setup requirements are manageable
  - Monitoring Management is easily accessible…
  - … but requires knowledge about program internals

- **Setup**

- **Results & Discussion**

- **Efficiency**
  - Faster execution of repeating tasks

- **Memorability**
  - Workflow seemed to be memorable …
  - … but casual users might show another result

- **Low error rate**
  - System helped to resolve errors during operation…
  - … but did not prevent flaws

- **Satisfaction**
  - Graph design is pleasant to use
  - Management dialogs require more work
Conclusions & Future Work

- Observations indicate good usability
- Simple centralized monitoring management
- Application Discovery method works

More research required
Feature enhancements
But what about JKD 9?