Visualization of OSGi based Software Architectures in Virtual Reality

Lisa Nafeie

Intelligent and Distributed Systems, German Aerospace Center (DLR) & Technische Hochschule Köln Cologne, Germany
Introduction

Intelligent and Distributed Systems
German Aerospace Center (DLR)

University of Applied Sciences
Cologne, Germany

@LisaNafeie
Pointing out Problems

- Visualization of Software Architectures: Large projects can become confusing
- Complex Systems: Uncluttered
- 2D Visualization: Less information

Software Dependencies
Software Projects in Space and Aerospace
Research Topics – DLR Intelligent and Distributed Systems

- Distributed Systems
- Workflows & Provenance
- Blockchains
- Machine Learning
- Software Engineering
- Software Analytics
Distributed Integration System

Remote Component Environment

- Language: Java
- Platform: RCP
- Framework: OSGi
Software Engineering

Focus on..

• Automated Testing

• Repository Mining

• Integration
Software Analytics

Definition

- Human-machine-interface
- Insightful & actionable information
- Tasks
  - software development
  - Systems
  - Users
Software Visualization
Interactive Visualization of OSGi-based Software Architectures

Goals

• Getting an impression of the dimensions of the application

• Introducing a new member of the development team

• Checking for abnormalities in the architecture
Java Framework: OSGi

Components & Dependencies

- Bundle
- Package
- Service
- Class

Source: D. Seider, A. Schreiber, T. Marquardt and M. Brüggemann, "Visualizing Modules and Dependencies of OSGi-Based Applications," 2016 IEEE Working Conference on Software Visualization (VISSOFT), Raleigh, NC, 2016, pp. 96-100.
Software Analytics – Source Code Analysis and Repository Mining

Repository

git

OSGi Application

Java
XML
.MF

Data Mining

Scanning and Analysis
JQAssistant

Graph Database
Neo4j

Visualization

VR IslandViz
Unity3D / C#

Devices
Oculus Rift / HTC Vive
Visualization in 2D: Dependencies of Key Bundles

Pro

- Bundle size visualized

Cons

- Less information of all components
  - Bundle name
  - Packages name
  - Class name

- Only 2 components visualized

Source: D. Seider, A. Schreiber, T. Marquardt and M. Brüggemann, “Visualizing Modules and Dependencies of OSGi-Based Applications,” 2016 IEEE Working Conference on Software Visualization (VISSOFT), Raleigh, NC, 2016, pp. 96-100.
Visualization in 2D: Package Structure of a Bundle

**Pro**
- Packages size visualized
- Class size visualized

**Cons**
- Less information of all components
  - Bundle name
  - Packages name
  - Class name
- Missing Relationships
  - Import & Export of Packages
Visualization in 2D: Dependencies Between Classes

**Pro**
- Shows Bundle Dependencies

**Cons**
- Missing Dependencies between & inside packages
- Confusing for large projects
Visualization in VR:
Prototype App “OSGiViewer”

**Pro**
- Third dimension
- Shows dependencies

**Cons**
- Less information of all components
- Packages name
- Class name
Technology

Microsoft HoloLens

HTC Vive

Oculus Rift

Tobii Eye Tracking
Virtual Reality & Augmented Reality
Virtual Reality App – “Dependency Diver“

Software Architecture

- OSGi Framework
- RCE Project
- Android App (Beta version)
- Motion Sickness
VR Approach 1: Module Stacks

VR Approach 1: Prototype “Dependency Diver“

Components

• need constant positions
• need a realistic metaphor

… to reduce **Motion Sickness**
VR Approach 1: Prototype “Dependency Diver“

Components include

• Packages
• Classes
VR Approach 1: Prototype “Dependency Diver“

Pro

✅ Cheap Headsets
✅ Interactive App

Cons

❌ Motion Sickness
VR Approach 2: „IslandViz“
Virtual Table
Navigation
Virtual Tablet
Island Metaphor

Classes
Multi-storey buildings with a new storey for every n lines of codes

Packages
Continuous regions

Bundles
Islands with multiple regions; each island with distinct shape
Dependencies

Ports
Incoming and outgoing package dependencies

Arrows
Strength and direction of a package dependency

Services
OSGi service interfaces and service components
Services
Augmented Reality
Positioning virtual surface on any surface
Visualization of OSGi based Software Architectures in Virtual Reality
### Approach 1: Dependency Diver

<table>
<thead>
<tr>
<th>Pro</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cheap Headsets</td>
<td>Limited usage time</td>
</tr>
<tr>
<td>Interactive App</td>
<td>Motionsickness</td>
</tr>
</tbody>
</table>

### Approach 2: IslandViz

<table>
<thead>
<tr>
<th>Pro</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Any (Cheap) Headsets&quot;</td>
<td>Currently available Java &amp; OSGi</td>
</tr>
<tr>
<td>Real-world Metaphor</td>
<td></td>
</tr>
<tr>
<td>Interactive App</td>
<td></td>
</tr>
<tr>
<td>Open Source</td>
<td></td>
</tr>
<tr>
<td>Detailed information of OSGi Components</td>
<td></td>
</tr>
</tbody>
</table>
Approach 1: Dependency Diver

Google PlayStore: “DependencyDiver”

Approach 2: IslandViz

https://github.com/DLR-SC/island-viz
Links

GitHub "IslandViz" - https://github.com/DLR-SC/island-viz


jQAssistant - https://jqassistant.org/

RCE - http://rcenvironment.de/


2D – 3D? - https://elib.dlr.de/110129/1/vissoft-toolpaper-osgivis_SeiderEtAl.pdf

DLR Jobs - https://www.dlr.de/dlr/jobs/#S:479