Architectural Visualization

Team Members: Mukesh Dangi, Barkha Bhojak, Supicha Phadungsilp, Umamahesh Emani

Aim

- Different team members, purposes and agendas.
- Abstract architecture is important.
- Levels and hierarchy can help separate concerns.
- Vulnerabilities in software need to be highlighted and depicted.

Input and Tools

Input:

- RELAX
  - deps.rsf
  - Relax_clusers_fn.rsf
- Source Code

Tools:

- D3.js - v4.0
- Apache Maven
- Java
- Python
- Express/Node.js

Level 1

- Provides an overall picture of the system.
- Bubble visualization helps users comprehend where a majority of the functionality of the system lies.
- Thick lines and arrowheads indicate a higher coupling and dependency.
- Useful for stakeholders and non-technical users.

Level 2

- Architect is responsible for designing, planning and (sometimes) developing the system.
- Descriptive and prescriptive architecture varies as the software increases in size.
- Architectural decay and erosion needs to be identified.

Level 2 (cont)

- Files are grouped according to their RELAX categories and sub-grouped by paths to the files/directory structure.
- Circle radius depends on the file size.
- Incoming and outgoing dependencies are shown as individual class files.
- The color of the box depicts the category the file belongs to.

Level 3

- Provides implementation level details for each file in the system.
- Details include lines of code, size of files, and dependencies.
- Useful for the developer in particular.

Word Cloud

- Additional information provided to the team.
- Displays a maximum of 30 files that have a large number of outgoing dependencies. The input can be varied.
- Can reinforce information about the prescriptive architecture of the system or show decay.

Thank you.