

Evaluating code complexity of SPL variability management:

Wrapper from pure::variants:

```
1 //PV: IFCOND (pv:hasFeature(HazardWarning))
2 static int warning_lights_value; [REMAINING CODE OF HazardWarning]
3 //PV: ENDCOND
```

Adapted to code

Emergence of Code Unpluggable Decorators-Based Annotations

Our decorator:

```
3 @@{"computerOpponent": "true"}
4 public class ComputerPlayer extends AbstractPlayer{
```

Adapted to code

```
import { PuzzleManagerService } from "src/app/services/puzzleGenerator/puzzle-manager.service";
var EXPRESSION_START91 = { "zoomCoordinates": "true" };
import { SetZoomComponent } from ".set-zoom/set-zoom.component";
var EXPRESSION_END91 = { "EXPRESSION_END": "-----" };
[ ... OTHER IMPORTS ... ]
@Component({
  selector: "app-zoom-management", templateUrl: ".zoom-management.component.html",
  styleUrls: [".zoom-management.component.scss"]
})
export class ZoomManagementComponent implements ZoomManagementInterface, AfterViewInit {
  centerX = 25;
  [ ... OTHER CLASS VARIABLES ... ]
  constructor(puzzleManagerService: PuzzleManagerService,
    @DecoratorTypesService.skipLineParameter({ "OR": { "zoomCoordinates": "true", "zoomValue": "true" } })
    componentFactoryService: ComponentFactoryService) {
    this.zoomManagerService = puzzleManagerService.getZoomManagerService();
    this.zoomManagerService.initForComponent(this);
  }
  ngAfterViewInit(): void { this.getComponentElement(this); }
  EXPRESSION_START97 = { "zoomCoordinates": "true" };
  public addZoomCoordinates(zoomHTML: Element | null): void {
    const positionAfterHeading = function <T>(domLocation: HTMLElement, componentRef: ComponentRef<T>) {
      domLocation.insertBefore(componentRef.location.nativeElement, domLocation.children.item(1));
      this.instantiateZoomCoordinatesComponentAndEvents(zoomHTML, this, positionAfterHeading);
    }
  }
  EXPRESSION_END97 = { "EXPRESSION_END": "-----" };
  public getComponentElement(self: ZoomManagementComponent): Element | null | undefined {
    const zoomHTML = document.getElementsByClassName("zoom-management-part").item(0);
    var EXPRESSION_START93 = { "zoomValue": "true" };
    this.addZoomValue(zoomHTML);
    var EXPRESSION_END93 = { "EXPRESSION_END": "-----" };
    [ ... OTHER CALLS ... ]
    return zoomHTML;
  }
  [ ... OTHER METHODS ... ]
  [ ... OTHER CONTENT ... ]
```

```
import { PuzzleManagerService } from "src/app/services/puzzleGenerator/puzzle-manager.service";
var EXPRESSION_START101 = { "zoomCoordinates": "true" };
import { SetZoomComponent } from ".set-zoom/set-zoom.component";
var EXPRESSION_END101 = { "EXPRESSION_END": "-----" };
[ ... OTHER IMPORTS ... ]
@Component({
  selector: "app-zoom-management", templateUrl: ".zoom-management.component.html",
  styleUrls: [".zoom-management.component.scss"]
})
export class ZoomManagementComponent implements ZoomManagementInterface, AfterViewInit {
  centerX = 25;
  [ ... OTHER CLASS VARIABLES ... ]
  constructor(puzzleManagerService: PuzzleManagerService,
    @DecoratorTypesService.skipLineParameter({ "OR": { "zoomCoordinates": "true", "zoomValue": "true" } })
    componentFactoryService: ComponentFactoryService) {
    this.zoomManagerService = puzzleManagerService.getZoomManagerService();
    this.zoomManagerService.initForComponent(this);
  }
  ngAfterViewInit(): void { this.getComponentElement(this); }
  @DecoratorTypesService.wholeBlockMethod({ "zoomValue": "true" })
  public addZoomValue(zoomHTML: Element | null): void {
    const positionAfterHeading = function <T>(domLocation: HTMLElement, componentRef: ComponentRef<T>) {
      domLocation.insertBefore(componentRef.location.nativeElement, domLocation.children.item(1));
      this.instantiateZoomValueComponentAndEvents(zoomHTML, this, positionAfterHeading);
    }
  }
  public getComponentElement(self: ZoomManagementComponent): Element | null | undefined {
    const zoomHTML = document.getElementsByClassName("zoom-management-part").item(0);
    var EXPRESSION_START103 = { "zoomValue": "true" };
    this.addZoomValue(zoomHTML);
    var EXPRESSION_END103 = { "EXPRESSION_END": "-----" };
    [ ... OTHER CALLS ... ]
    return zoomHTML;
  }
  [ ... OTHER METHODS ... ]
```

G	H	I	J
Halstead Difficulty	Halstead Effort	Halstead Length	Halstead Time
8.13	3060.883	74	170.049
5.893	2149.082	69	119.393
14.211	19460.57	216	1081.143
8.069	3717.772	86	206.543
14.737	18517.35	203	1028.8
4.857	758.103	35	42.117
4.75	903.429	40	50.19
14.537	33312.59	348	1850.7
5.125	2448.68	87	136.038
11.217	15945.04	247	885.836
3.5	239.863	18	13.326
12	20255.46	267	1125.303
11.684	15559.4	267	1125.303
2.5	340.0	18	13.326
2.708	232.4	18	13.326
2.679	261.0	18	13.326
2.5	148.0	18	13.326
2.667	276.0	18	13.326
2.5	161.0	18	13.326
2.5	161.0	18	13.326
4	903.0	18	13.326
21.309	37671.0	18	13.326

Cf1: SOURCE CODE IN FORM 1

Cf2: SOURCE CODE IN FORM 2

METRICS OF Cf1

METRICS OF Cf2

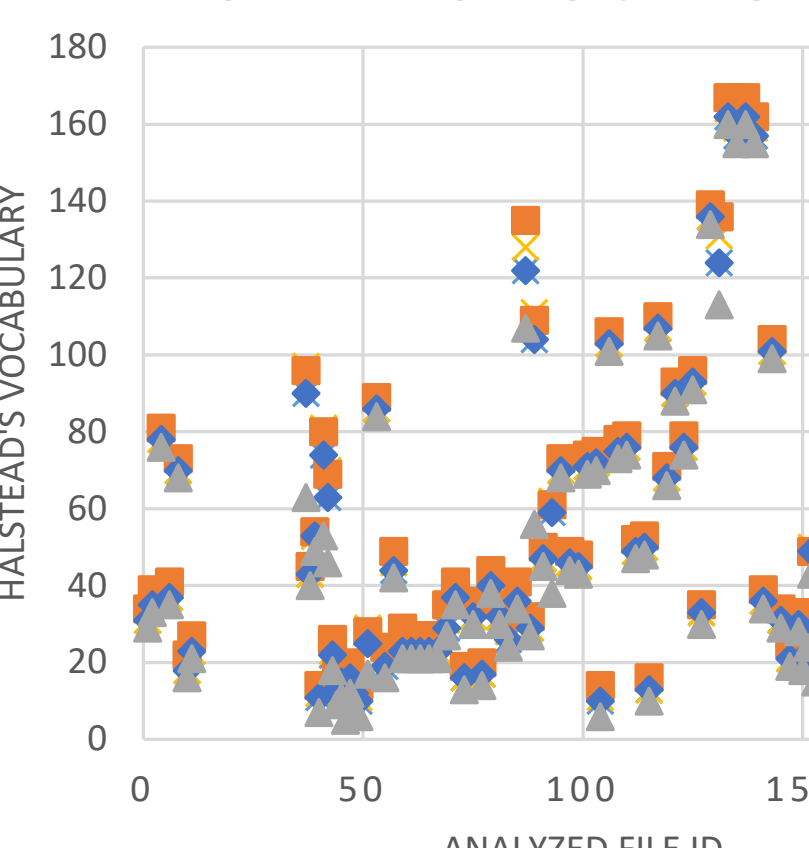
TESTING SIGNIFICANCE OF DIFFERENCE

DIFFERENCES BETWEEN Cf1 AND Cf2 METRICS

Name of compared metric	Corr.	W	p-value	95% CI	Est.	p>0.05
Cyclomatic Complexity	1.0000	0	1.0000E+00	NaN, NaN	NaN	TRUE
Cyclomatic Density	0.8226	0	3.5776E-13	-4.1959, -2.28	-3.02	FALSE
Halstead's Bugs	0.9997	2556	2.4526E-13	0.01, 0.02	0.0141	FALSE
Halstead's Difficulty	0.9971	2237	5.9298E-09	0.60, 0.80	0.7390	FALSE
Halstead's Effort	0.9988	2386	2.2106E-10	503.04, 1493.10	841.6983	FALSE
Halstead's Length	0.9997	2556	9.3382E-17	6.00, 6.00	6.0000	FALSE
Halstead's Time	0.9988	2386	2.2106E-10	27.95, 82.95	46.7609	FALSE
Halstead's Vocabulary	0.9994	2484	4.2116E-14	3.00, 3.45	3.0000	FALSE
Halstead's Volume	0.9997	2556	2.4761E-13	39.32, 45.96	42.2363	FALSE
Halstead's Id Dist. Operands	0.9996	2415	2.5713E-16	2.00, 2.00	2.0000	FALSE
Halstead's Id Ttl Operands	0.9999	2556	9.3382E-17	2.00, 2.00	2.0000	FALSE
Halstead's Id Dist. Operators	0.9886	2030	2.1410E-12	1.00, 1.50	1.0000	FALSE
Halstead's Id Ttl Operators	0.9999	2485	9.8502E-17	4.00, 4.00	4.0000	FALSE
LOC Physical	0.9998	2556	2.1563E-16	1.00, 1.00	1.0000	FALSE
LOC Logical	0.9999	2485	9.8502E-17	2.00, 2.00	2.0000	FALSE

Used metrics

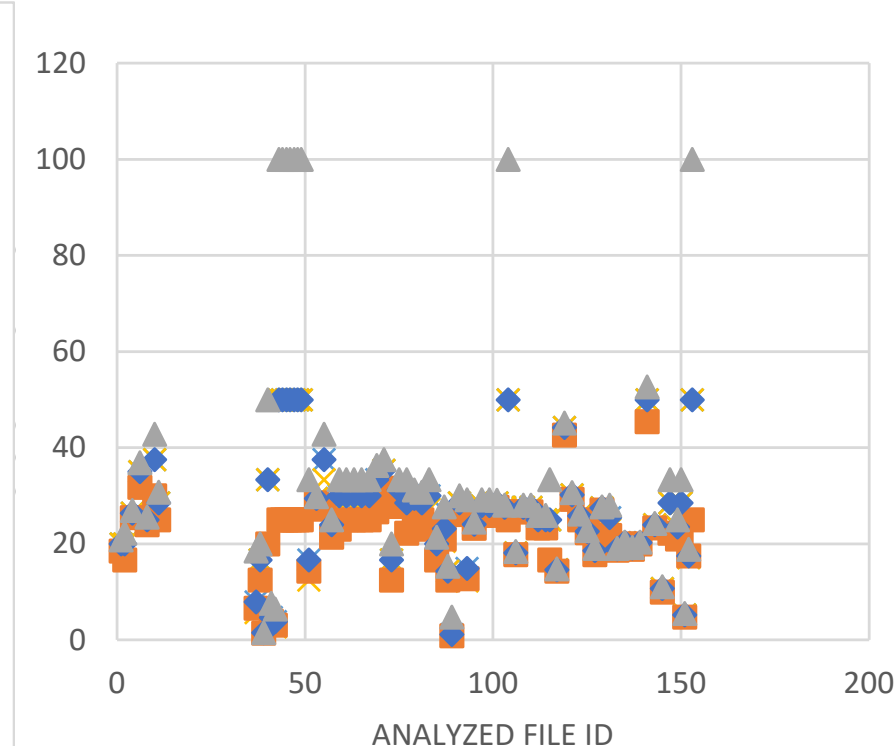
HALSTEAD'S VOCABULARY



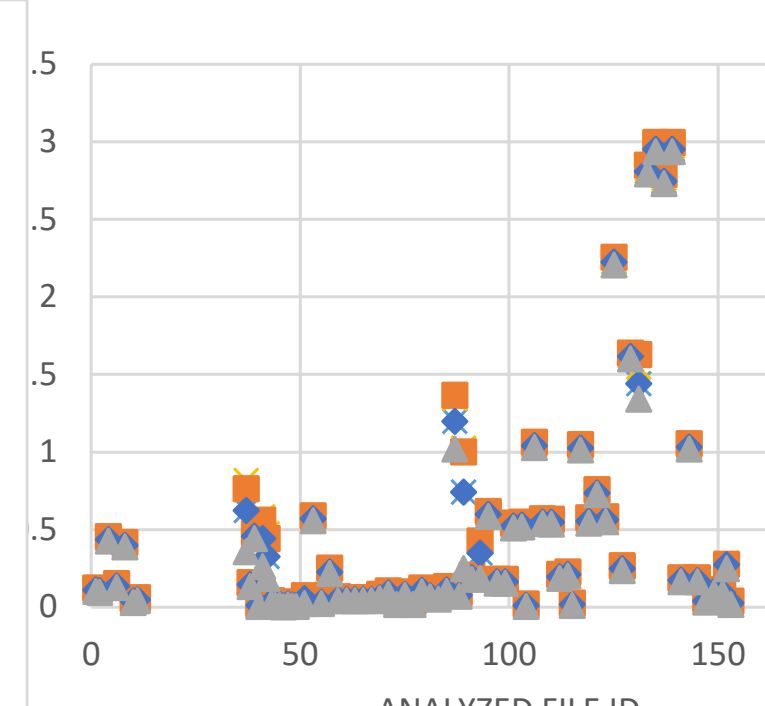
Comparative analysis of:

Visualization

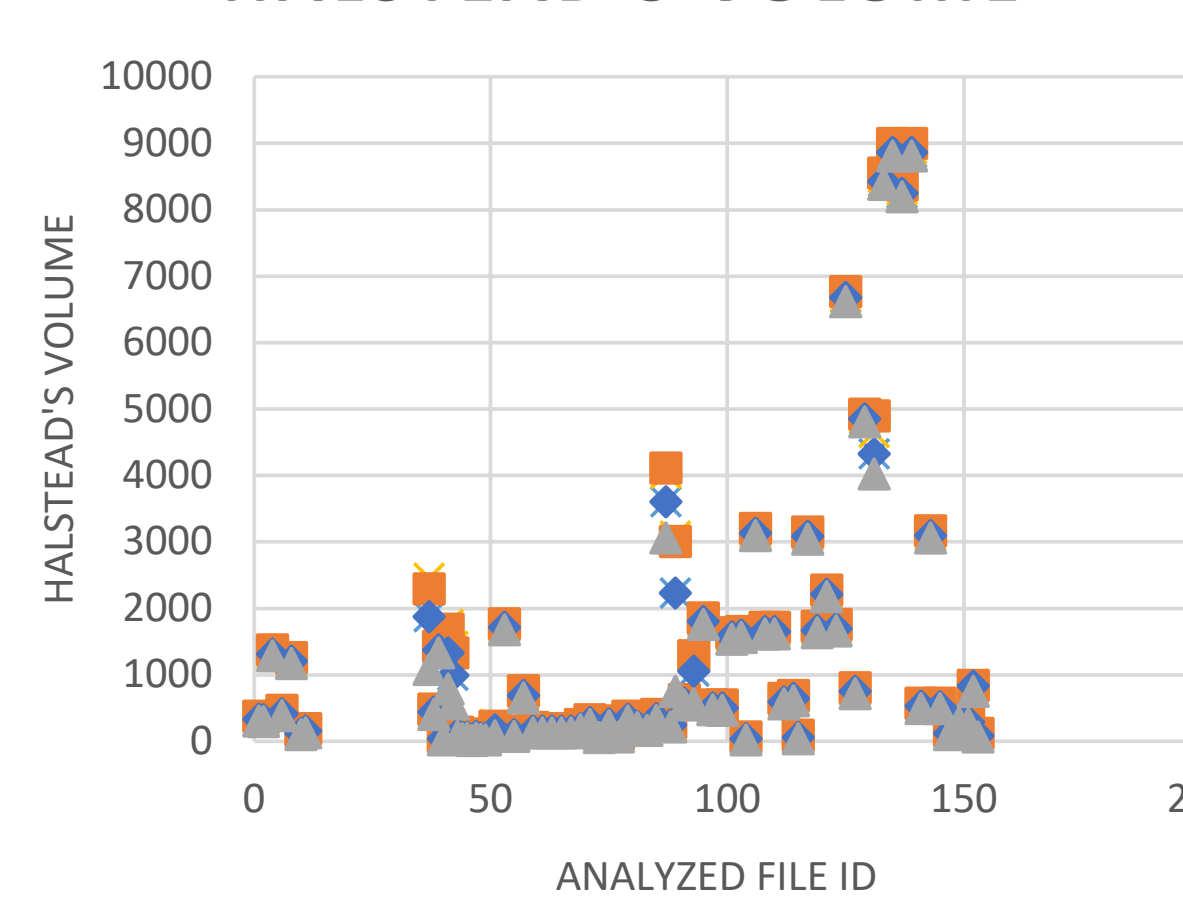
CYCLOMATIC DENSITY



HALSTEAD'S BUGS



HALSTEAD'S VOLUME



Link

Code: <https://github.com/jperdek/variabilityMgmtCodeConstructsComplexity>

Author:

Ing. Jakub Perdek

perdek.jakub@gmail.com

Leader: Prof. Ing. Valentino Vranić Phd.

vranic@stuba.sk