DO-332 OOP Verification User Guide

Overview

This guide explains how to use the STUNIR DO-332 OOP verification tools to analyze object-oriented code for DO-178C compliance.

Installation

Prerequisites

- GNAT Ada compiler (2022 or later)
- GNATprove (for SPARK proofs)
- Make

Building

```
cd tools/do332
make build    # Debug build
make release  # Release build
make prove    # Run SPARK proofs
```

Basic Usage

Running Analysis

```
# Basic analysis
./bin/do332_analyzer

# Specify options
./bin/do332_analyzer --ir-dir asm/ir --dal B --verbose

# Show help
./bin/do332_analyzer --help
```

Output

The analyzer produces reports in JSON format:

- `inheritance_report.json` - Inheritance analysis
- `polymorphism_report.json` - Polymorphism verification
- `dispatch_report.json` - Dynamic dispatch analysis
- `coupling_report.json` - Coupling metrics
- `generated_tests.json` - Test cases

DAL Levels

The analyzer supports configurable analysis based on DAL level:

DAL	Inheritance	Polymorphism	Dispatch	Coupling
A	Full	Full + LSP	Full + Timing	Full
B	Full	Full + LSP	Full	Full
C	Full	Basic	-	Basic
D	Basic	Basic	-	-
E	-	-	-	-

Integration

With STUNIR Build

```
export STUNIR_ENABLE_COMPLIANCE=1
export STUNIR_ENABLE_DO332=1
./scripts/build.sh
```

With Python

```
from do332_wrapper import DO332Analyzer

analyzer = DO332Analyzer()
results = analyzer.analyze(dal="B")
```

Understanding Results

Inheritance Analysis

- **Depth**: Inheritance tree depth (threshold: 5-6)
- **Diamond**: Diamond inheritance patterns
- **Overrides**: Method override verification

Coupling Metrics

- **CBO**: Coupling Between Objects (< 14 recommended)
- **RFC**: Response For Class (< 50 recommended)
- **LCOM**: Lack of Cohesion in Methods
- **DIT**: Depth of Inheritance Tree
- **NOC**: Number of Children