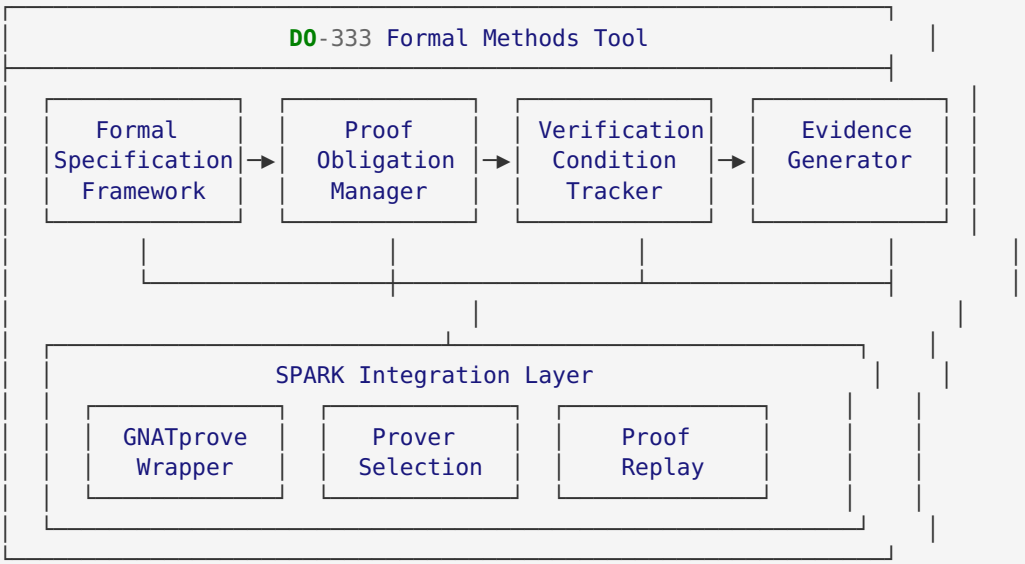
DO-333 Formal Methods Support - Architecture

Overview

The DO-333 module provides formal verification capabilities integrated with the STUNIR pipeline, supporting certification under DO-178C with DO-333 Formal Methods Supplement.

Component Architecture



Package Dependencies

```
do333_main.adb
    ├── formal_spec.ads/adb
    │       └── spec_parser.ads/adb
    ├── proof_obligation.ads/adb
    │       └── po_manager.ads/adb
    ├── verification_condition.ads/adb
    │       └── vc_tracker.ads/adb
    ├── spark_integration.ads/adb
    │       └── gnatprove_wrapper.ads/adb
    ├── evidence_generator.ads/adb
    └── report_formatter.ads/adb
```

Data Flow

```
Input Sources                Processing                    Outputs
═══════════════════════════════════════════════════════════════════

Ada/SPARK Source -----> Spec_Parser ----> Contract_Spec
                              |              |
                              v              v
STUNIR IR -----------> Formal Spec ----> PO_Collection
                              |              |
                              v              v
GNATprove Output ----> PO_Manager ----> VC_Collection
                              |              |
                              v              v
                       VC_Tracker ----> Evidence Reports
                              |              |
                              v              v
                       Evidence_Gen -> Compliance Matrix
```

Key Data Structures

Formal_Expression

```
type Formal_Expression is record
   Kind       : Spec_Kind;      -- Pre, Post, Invariant, etc.
   Content    : Expr_String;    -- Expression text
   Length     : Expr_Length;
   Line_Num   : Natural;
   Column     : Natural;
   Verified   : Boolean;        -- Has been proved
   Traceable  : Boolean;        -- Has requirement trace
end record;
```

Proof_Obligation_Record

```
type Proof_Obligation_Record is record
   ID           : Natural;
   Kind         : PO_Kind;           -- Precondition, Range_Check, etc.
   Status       : PO_Status;         -- Proved, Unproved, Timeout, etc.
   Criticality  : Criticality_Level; -- DAL-A through DAL-E
   Strategy     : Discharge_Strategy;
   Source_File  : Source_String;
   Subprogram   : Subp_String;
   Line, Column : Natural;
   Context      : Context_String;
   Prover_Time  : Natural;           -- milliseconds
   Step_Count   : Natural;
end record;
```

VC_Record

```
type VC_Record is record
   ID           : Natural;
   PO_ID        : Natural;           -- Parent proof obligation
   Status       : VC_Status;
   Complexity   : Complexity_Category;
   Formula      : Formula_String;
   Step_Count   : Natural;
   Time_MS      : Natural;
   Prover       : Prover_String;
end record;
```

SPARK Mode

All packages have `SPARK_Mode (On)` with:
- Formal contracts (Pre/Post)
- No runtime exceptions
- Flow analysis clean
- Automatic proof where possible

Design Decisions

1. **Bounded Types**: All strings/arrays use fixed bounds for SPARK
2. **Explicit State**: No global state, all data passed explicitly
3. **Defensive**: Overflow checks, range validation throughout
4. **Deterministic**: Same inputs produce same outputs
5. **Self-Verifying**: Tools prove their own correctness

Integration Points

With DO-331 (SysML 2.0)

- Import model elements for formal specification
- Trace specifications to requirements

With DO-332 (OOP)

- Verify Liskov Substitution Principle
- Prove dispatching correctness

With STUNIR Pipeline

- Optional feature via `STUNIR_ENABLE_COMPLIANCE`
- No breaking changes to existing pipeline