Stood and the AADL

AADL Workshop
Paris, 17-18 October 2005
about TNI Europe

Company created in year 2000
Headquarters in the UK, offices in F (hosted by Ellidiss Technologies)
Former subsidiary of TNI Valiosys (independant since Jan 2004)
Edits, distributes and supports CASE tools: CP-Hood and Stood
Involved in many avionics, space, military and ground transportation
critical SW development, including Eurofighter, A380
about Stood

- multi-standard software modeling tool:
  - UML2.0, HOOD 4.0, HRT-HOOD and AADL 1.0

- well defined step-by-step modeling process:
  - import of functional and non-functional requirements
  - multi-notation graphical design of the architecture
  - multi-language detailed design and coding (Ada, C, C++, ...)
  - static design verification tools
  - multi-format documentation generators (html, pdf, word, ps, ...)
  - code generators and reverse engineering (i.e. Ada <-> AADL)

- ready for industrial projects:
  - multi-users, configuration management, requirements traceability
  - Unix-Windows interoperability
  - Complies with process standards:
    DO-178B, EN-50128, ECSS-E40, MIL-STD-498, ...
Stood/AADL history

- 2002-2003: the Cotre project
  - with Airbus, Féria, EnstB
- 2003-2004: mapping meta-models
  - (HRT-)HOOD, UML 2.0 and AADL components
- 2004-2005: transformation rules
  - AADL 1.0 input and output features
- ongoing work:
  - AADL graphical editor
  - tool coupling: Osate, Cheddar, Topcased, ...
  - supporting future annexes: Cotre (again), Assert, ...
  - AADL legality rules checker
Use the System design model as a blueprint for SW architectural design
import sw requirements

import system specs (AADL)

edit system specs (AADL)

blue print

software architecture (UML-HOOD)

software detailed design (HOOD)

design verification

automatic code generation

design documentation

application

spec

specs

repositories

reuse components

reuse legacy code

repository
example of
AADL specification

Preserve as much as possible
specified system architecture
example (cont.):
corresponding
software architectural model

Now ready for
detailed design activities,

code generation, ...
Stood overview

components

lifecycle

tabs

notation

switch

graphical

textual

editor

views

features and properties
AADL generator

open Osate

AADL syntactic analysis

AADL generated code

AADL view

1. (design) GPS
2. (design) mh_types
3. (design) mh_types_sonar_types
4. (design) mine_field
5. (design) mine_hunter
   - mine_hunter
   - mine_field
   - on_shore_center
   - GPS
   - mh_types
   - sonar_types
6. (design) on_shore_center
   - generic antenna

SYSTEM mine_hunting
FEATURES
start_mission : IN EVENT PORT;
END mine_hunting;

SYSTEM IMPLEMENTATION mine_hunting,others;

SUBCOMPONENTS
mine_hunter : PROCESS mine_hunter;
mine_field : PROCESS mine_field;
on_shore_center : PROCESS on_shore_center;
GPS : PROCESS GPS;

CONNECTIONS
EVENT PORT start_mission -> mine_hunter.start_mission;
DATA PORT mine_field.reflection -> mine_hunter.reflection;
DATA PORT mine_field.reverberation -> mine_hunter.reverberation;
DATA PORT mine_field.noises -> mine_hunter.noises;
DATA PORT GPS.global_pose -> mine_hunter.global_pose;
DATA PORT mine_hunter.signal -> mine_field.signal;
END mine_hunting,others;

-- The mine hunter operates on a mine field to detect, localize and classify
-- underwater mines.
-- Identified targets are reported to the on shore center.
PROCESS mine_hunter
FEATURES
    global_pose : IN DATA PORT mh_types::position;
signal : OUT DATA PORT mh_types::acoustic_signal;
reflection : IN DATA PORT mh_types::acoustic_signal;
reverberation : IN DATA PORT mh_types::acoustic_signal;
noises : IN DATA PORT mh_types::acoustic_signal;
targets : OUT DATA PORT mh_types::Mine_Descriptor;
start_mission : IN EVENT PORT;
END mine_hunter;

PROCESS IMPLEMENTATION mine_hunter,others

END_OBJ

-- The mine hunter operates on a mine field to detect, localize and classify
-- underwater mines.
-- Identified targets are reported to the on shore center.
PROCESS mine_hunter
FEATURES
    global_pose : IN DATA PORT mh_types::position;
signal : OUT DATA PORT mh_types::acoustic_signal;
reflection : IN DATA PORT mh_types::acoustic_signal;
reverberation : IN DATA PORT mh_types::acoustic_signal;
noises : IN DATA PORT mh_types::acoustic_signal;
targets : OUT DATA PORT mh_types::Mine_Descriptor;
start_mission : IN EVENT PORT;
END mine_hunter;

PROCESS IMPLEMENTATION mine_hunter,others
Conclusion

• Stood/AADL usage:
  – import AADL specs as blueprint for SW dev process
  – edit an AADL model and convert it into a SW design
  – export AADL specs from SW design or legacy code

• All the other SW design features are maintained:
  – requirements traceability
  – incremental documentation
  – design verification
  – automatic code and doc generation
  – round-trip and reverse engineering

• Industrial support available

download:
www.tni-world.com