

[www.tni-world.com](http://www.tni-world.com)

# *Stood and the AADL*

★ ★ **Tni** Europe  
**Technology and Innovation**

[www.tni-world.com](http://www.tni-world.com)

*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)

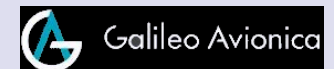
Former subsidiary of TNI Valiosys (independant since Jan 2004)

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**



TELDIX GmbH



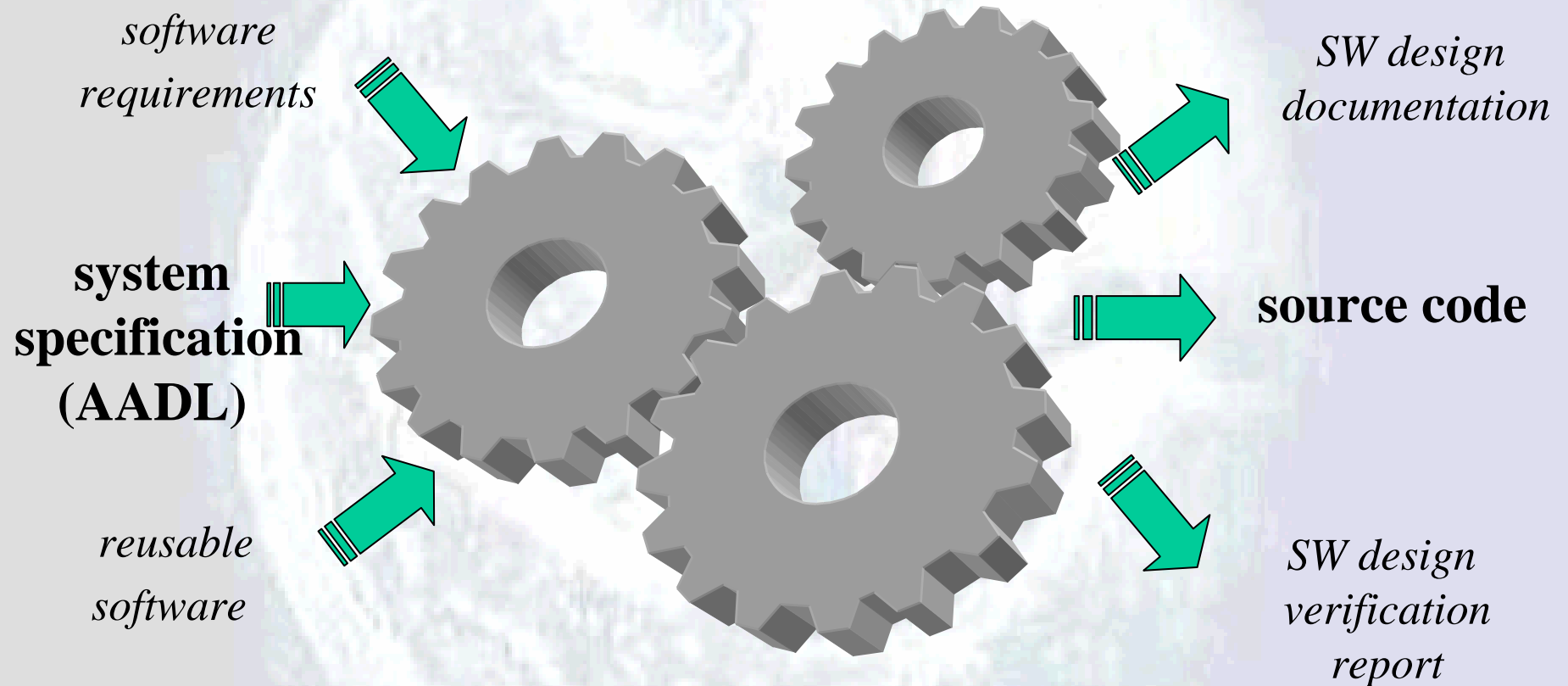
# about Stood

- 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

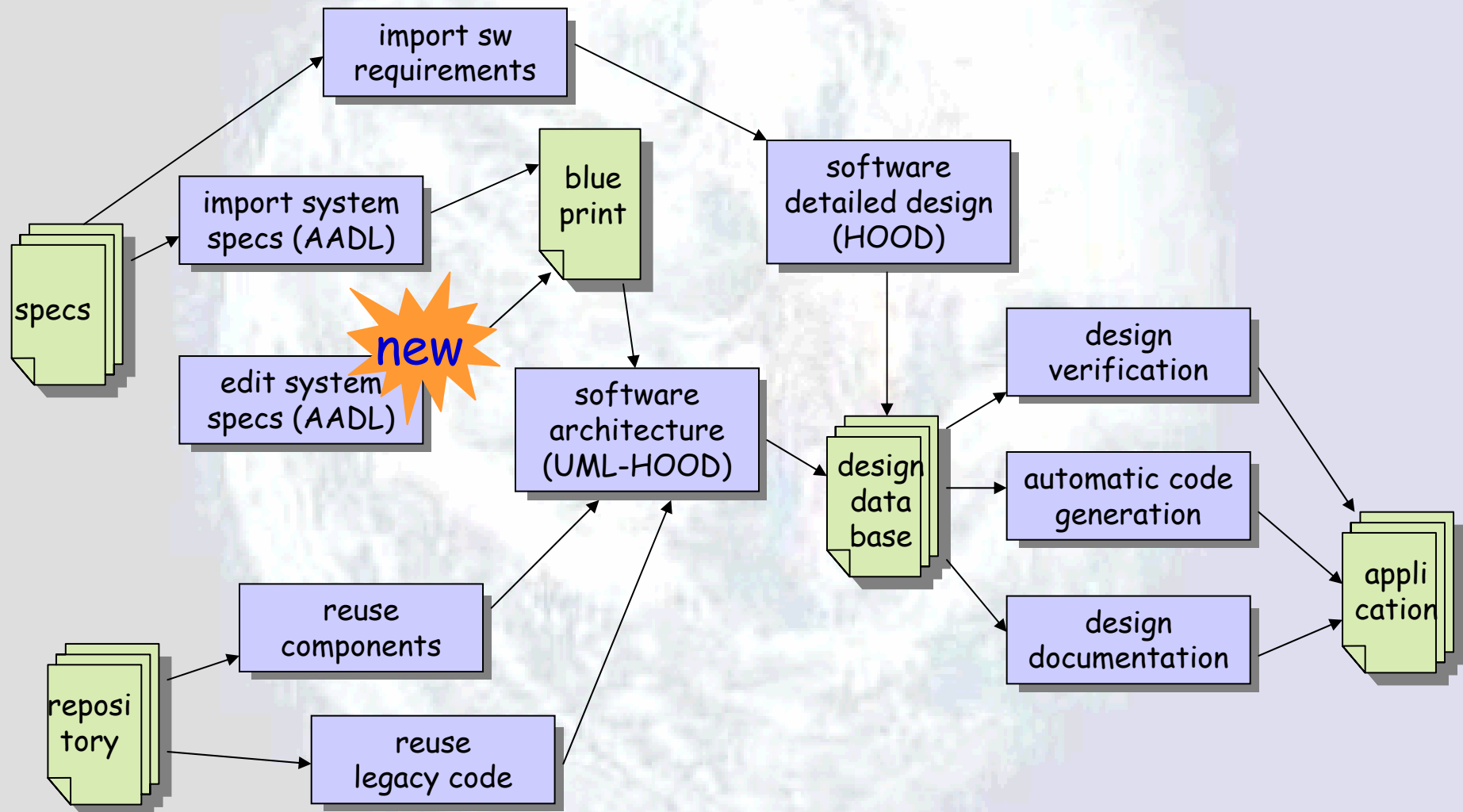
- 2002-2003: the Cotre project
  - with Airbus, Féria, EnstB
- 2003-2004: mapping meta-models
  - (HRT-)HOOD, UML 2.0 and AADL components
- 2004-2005: transformation rules
  - AADL 1.0 input and output features
- ongoing work:
  - AADL graphical editor
  - tool coupling: Osate, Cheddar, Topcased, ...
  - supporting future annexes: Cotre (again), Assert, ...
  - AADL legality rules checker

# Software production

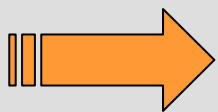
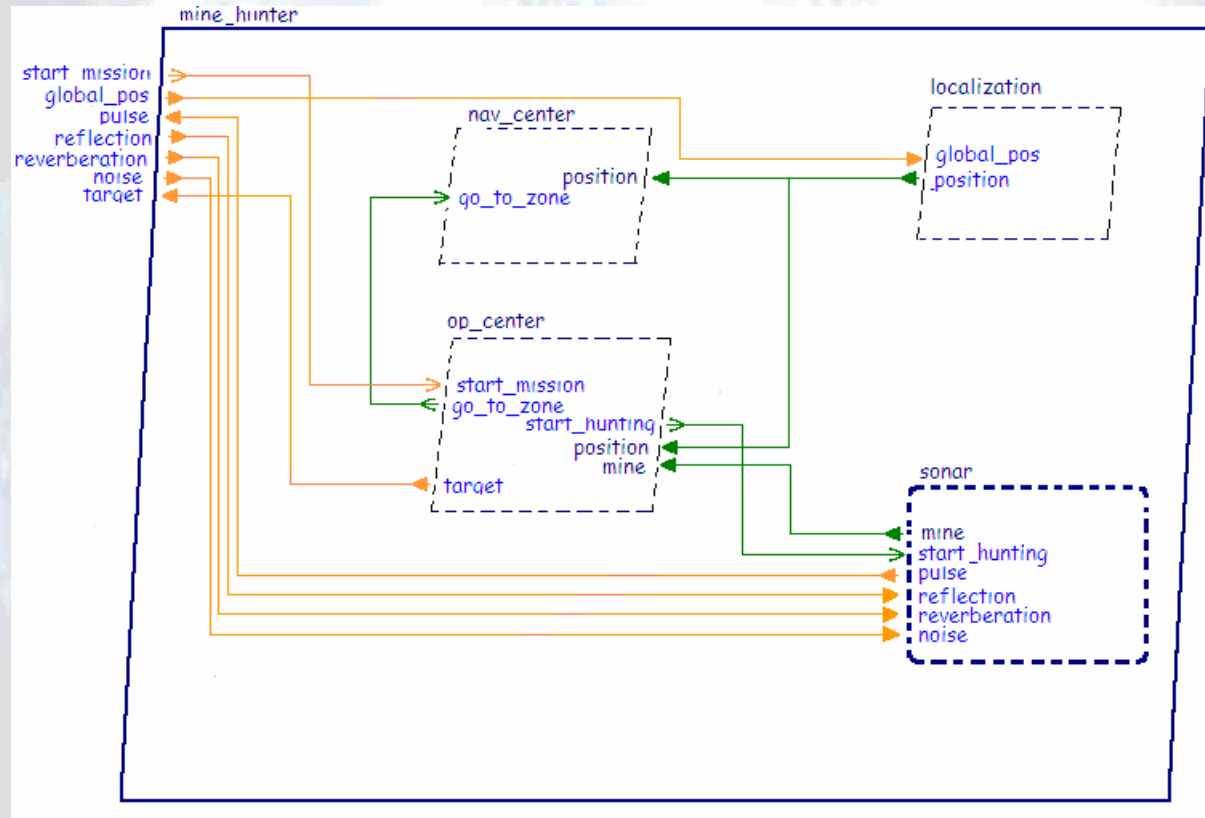


*Use the System design model  
as a blueprint for SW architectural design*

# Stood Workflow

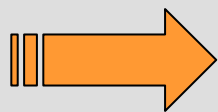
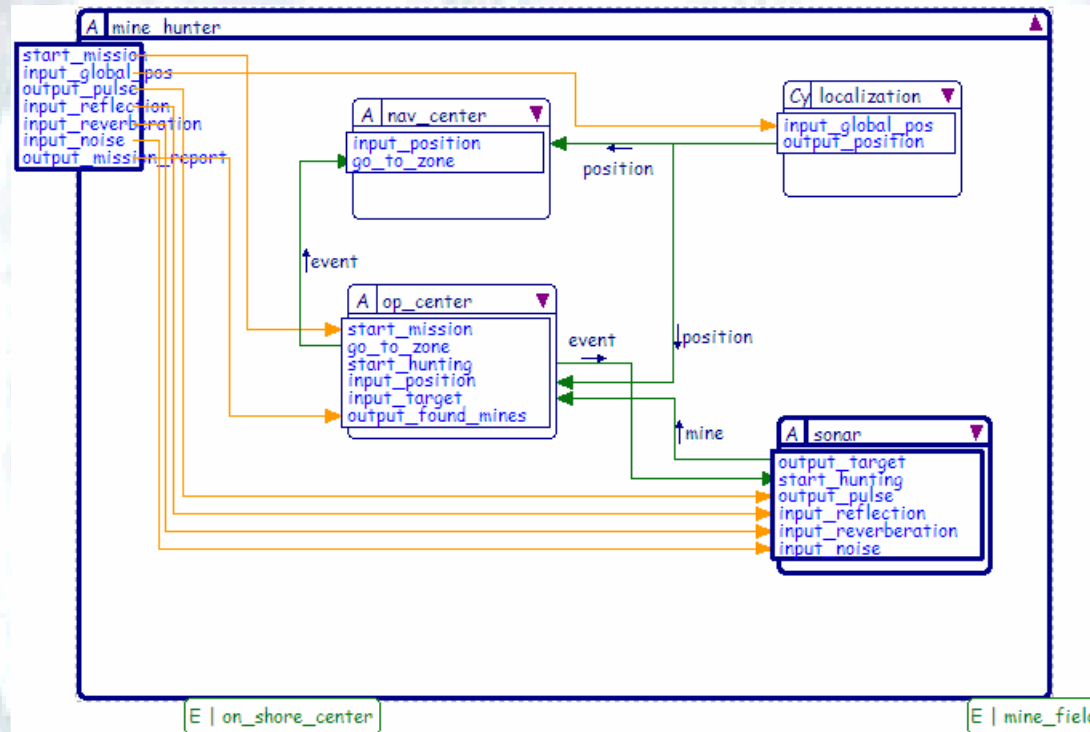


## example of AADL specification



*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

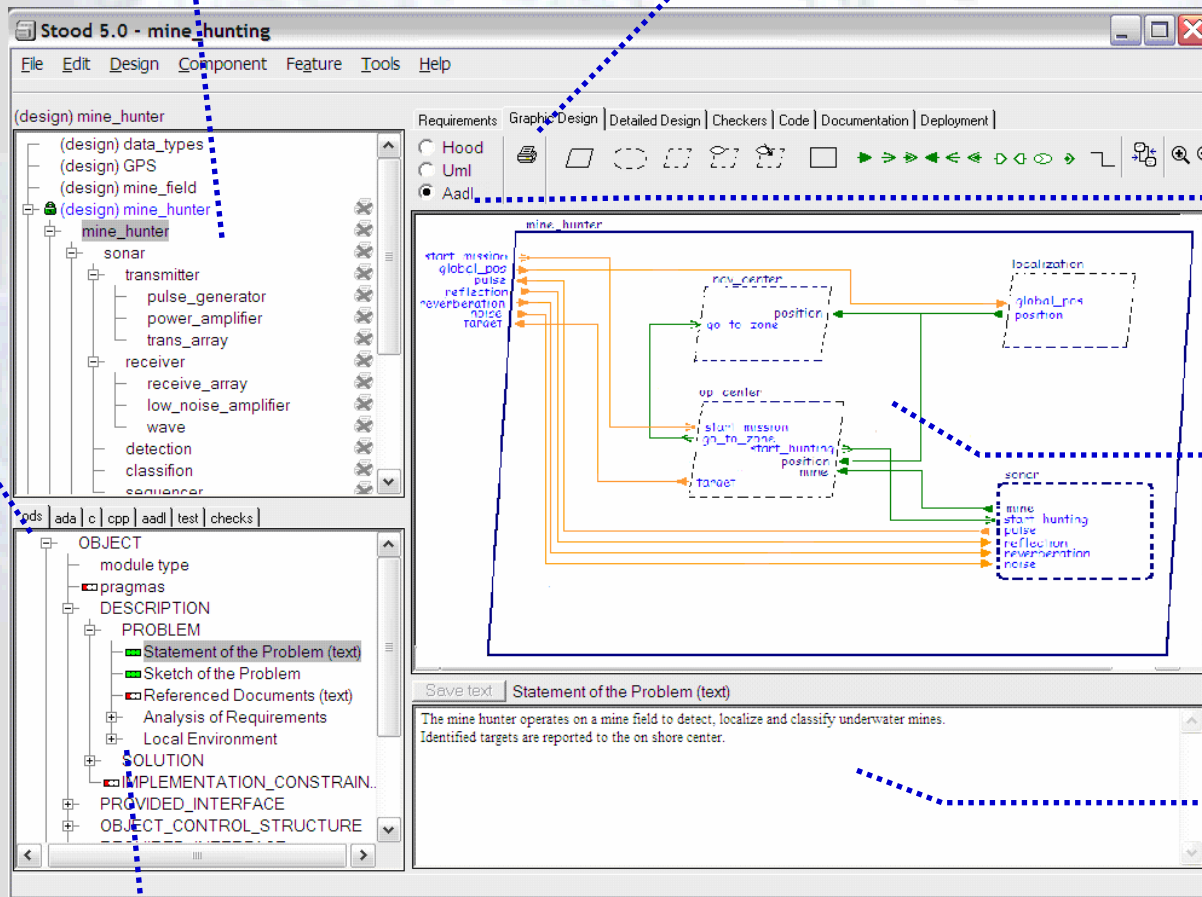
notation switch

graphical editor

textual editor

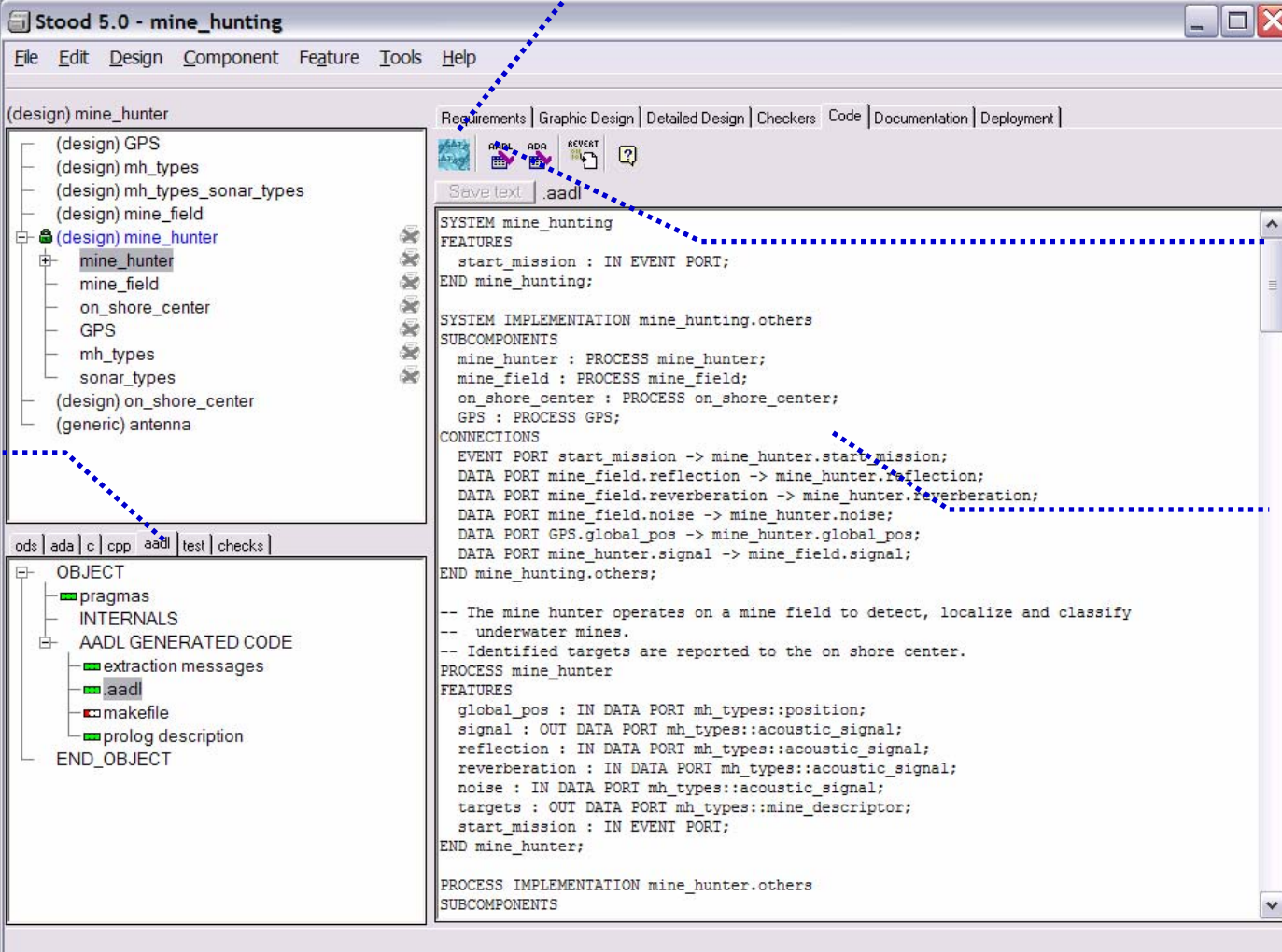
features and properties

views



# AADL generator

open  
Osate



The screenshot shows the Stood 5.0 - mine\_hunting application window. The interface includes a menu bar (File, Edit, Design, Component, Feature, Tools, Help), a toolbar with icons for AADL, ADA, REVERT, and HELP, and a 'Save text' button with a file extension of '.aadl'. The main workspace is divided into three panes:

- Left Pane (Design Tree):** Shows a hierarchical tree structure for '(design) mine\_hunter', including sub-designs like GPS, mh\_types, mine\_field, and mine\_hunter. The 'mine\_hunter' sub-design is expanded to show components like mine\_field, on\_shore\_center, GPS, mh\_types, and sonar\_types.
- Bottom-Left Pane (Object Browser):** Shows a tree structure for 'OBJECT', including 'pragmas', 'INTERNALS', 'AADL GENERATED CODE' (with sub-items like 'extraction messages', '.aadl', 'makefile', 'prolog description'), and 'END\_OBJECT'.
- Right Pane (Code Editor):** Displays the generated AADL code for 'SYSTEM mine\_hunting'. The code includes:
 

```

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.noise -> mine_hunter.noise;
  DATA PORT GPS.global_pos -> mine_hunter.global_pos;
  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_pos : 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;
  noise : 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
SUBCOMPONENTS
      
```

AADL  
view

AADL  
syntactic  
analysis

AADL  
generated  
code

# 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](http://www.tni-world.com)