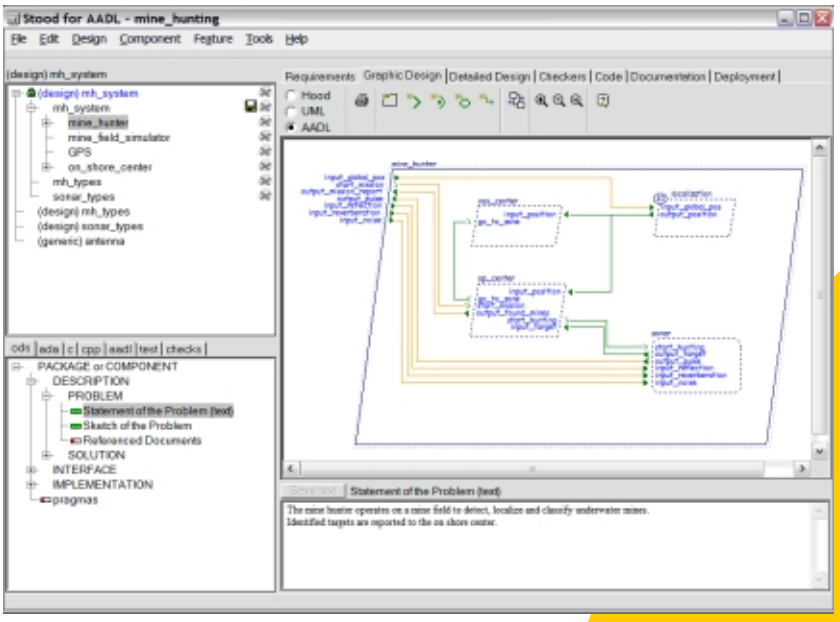
Features:



Project management

full Windows-Unix interoperability -
network distributed project bases -
integrated interface to remote Configuration Management Systems -
multi user management at system and subsystem level -
SIF and XML design model interchange -

Requirements traceability

- import of high level requirements
- incremental requirements coverage
- management of the derived requirements
- bidirectional interface with Reqtify™

Architectural Design

components based approach with black-box and white-box views -
UML 2.0 graphical notation -
AADL 1.0 graphical notation and import/export -
support of HOOD and HRT-HOOD methodology -
built-in real-time model -

Detailed Design & Coding

- customizable structured detailed design framework
- incremental documentation
- incremental coding and round-trip engineering
- incremental requirements coverage
- legacy Ada and C code reverse engineering

Verifications

cross references table -
automatic calculation of the required interfaces of the components -
automatic generation of call trees and dataflow graphs -
real-time schedulability analysis -
requirements traceability matrix -
design rules checker -
design metrics -

Code & Doc generators

- Ada95
- C/C++

- HTML
- PostScript/PDF
- RTF (Word™)
- MIF (FrameMaker™)

Compliancy to Standards

AS-5506: Architecture Analysis and Design Language
DO-178B for embedded avionics
ECSS-E40 for space systems
EN-50128 for railways
MIL-STD-498
...

Mountbatten Court
Worrall Street
Congleton
CW12 1DT
UK
+44 (0) 1260 291449

Ellidiss Software
TNI Europe Limited

Ellidiss Technologies
24 quai de la douane
29200 Brest

France
+33 (0) 298 451 870

download site:
www.ellidiss.com

Stood 5.2

AADL
System & Software
Modeling Tool




Better Quality and Higher Productivity

The increasing complexity of software is a key issue for developing high quality critical systems, while mastering costs and delays. Higher levels of Quality are expected to comply with the most demanding development processes and products certification, and Productivity gains must be achieved to cope with increasing competition pressure.

flexible Model Driven Engineering

Stood technology is based on advanced Model Transformations that can be easily customized or extented to new standards in order to optimize the integration of the tool within the existing development chain, and thus contribute actively to a powerful implementation of the Software production process.

Discover the 5 top reasons for choosing Stood 5 for your next project:

1 The right compromise between Maturity and Innovation

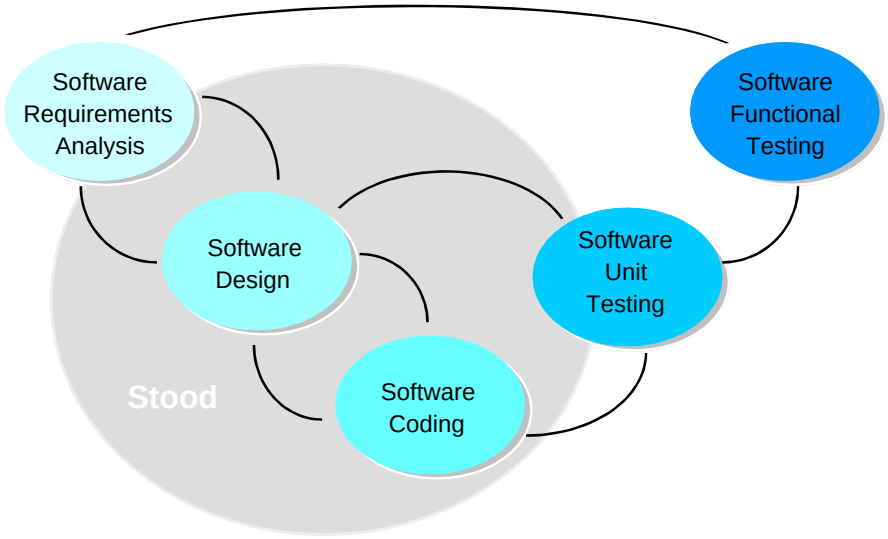
Innovation alone is not sufficient to solve the software crisis. No solution will bring the awaited benefits if it cannot be properly and smoothly integrated within the existing software development organizations, processes and teams. Continuous feed-back from long term, large scale and critical industrial projects must complement technical innovation to reach the qualified level of Maturity required by a state of the art solution.

Stood has been continuously improved by a well balanced set of technological innovations combined with day-to-day interactions with major industrial projects. It has become a seamless way to introduce new technology inside ongoing long term critical projects. The proven Maturity of the product minimizes the risks that are inherent in any technical Innovation. To enable existing projects to take full advantage of this 'State of the Art' technology Stood can import legacy design models and code.

2 Domain Specific instead of General Purpose

Many software modeling products consist more of a smart catalog of popular conceptual abstractions than of a true life-cycle support tool. Most of them tend to minimize the modeling constraints in order to maximize their marketing scope. For instance, a UML class diagram can be used to model any kind of software, but what does it add for a real-time embedded software?

Stood has been specifically developed to meet the industrial requirements provided by prime organizations like space agencies and major avionics firms. Its features have been tailored to offer the highest Value added for the restricted Scope of software architectural and detailed design activities. That's also why Stood includes a true built-in Real-Time model that is not just a late extension to a more general modeling technique. The underlying meta model of Stood describes a powerful Domain Specific Language for mission critical software modelling.



```
| tasks:              |   T   |   W   |   P   |   U   |   D   |   R
| hlw_handler         |  6000 |    20 |     6 |    0% |   200 |   6
| water_flow_sensor   |  1000 |    10 |     9 |    1% |    40 |   4
| ch4_sensor          |    80 |    12 |    10 |   15% |    30 |   1
| air_flow_sensor     |   100 |    10 |     7 |   10% |   100 |   4
| co_sensor           |   100 |    10 |     8 |   10% |    60 |   3
| summation:                                    36%
| bound:                                        74%
WARNING: Rate Monotonic Priority Ordering is not respected
INFORMATION: Deadline Monotonic Priority Ordering is respected
WARNING: Missed Dealine for: water_flow_sensor
| T: Period
| W: Worst Case Execution Time
| P: Priority
| U: Utilization
| D: Deadline
| R: Response Time
```

3 Correctness by construct and early Verification

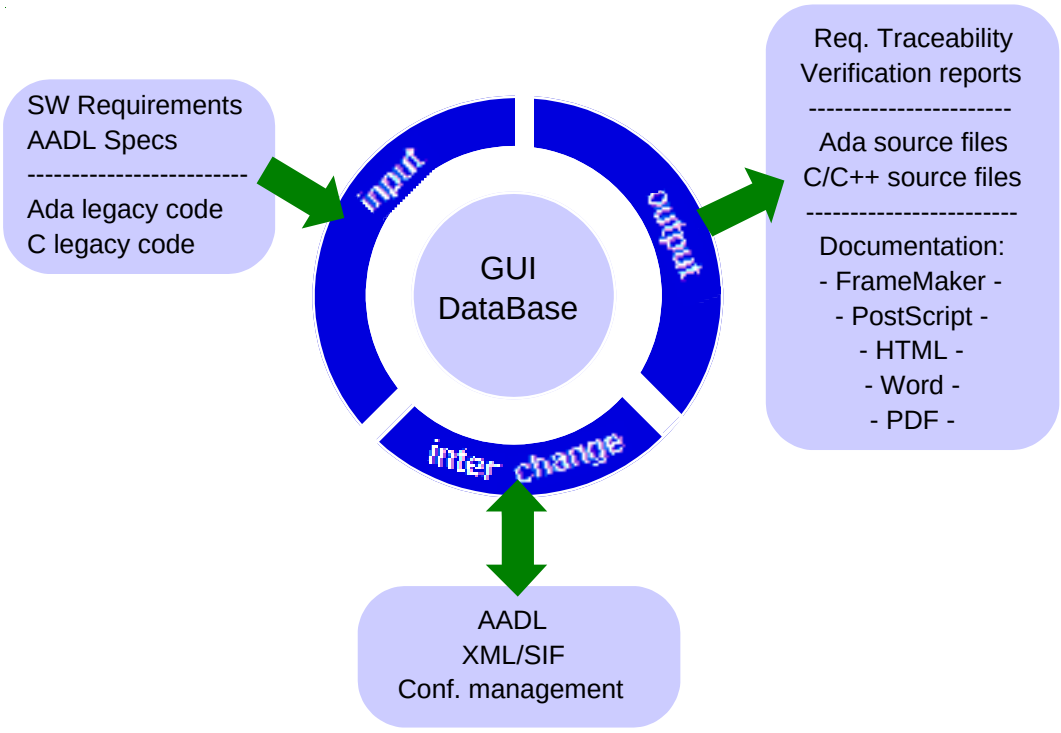
Every critical projects are developed under the constraint of rigorous design and coding rules that are required to comply with program or corporate wide quality standards or certification recommendations. The role of a software design tool is to support practical guidelines in order to cope with these constraints as early as possible in the modeling process. This aim can be achieved by offering a modeling process that leads to a software architecture that is Correct by Construct and a set of additional early model Verification tools at detailed design stage.

Stood enforces a hierarchical component based approach. The top-down software design process promotes well structured architectures of interacting components that will facilitate the subsequent software integration, testing and maintenance activities. A set of customizable rules verification tools empowered by Logical Model Processing (LMP) technology can also be used at any time to monitor key quality indicators like requirements coverage, schedulability, coupling and consistency between software design and code.

4 Reduce the gap between Software Design and Code

Many projects have suffered too much from the strong inconsistencies that can remain between the design documentation and the actual code of the application. This issue can be solved by considering the design model as the unique reference from which source code and documentation can be generated synchronously.

Stood detailed design database is structured to encompass all the necessary information to become a true reference for the project. Stood enforces incremental design documentation and coding all along the design activity. It offers automatic code generators and round-trip engineering where the design model to source code transformations are expressed by a set of customizable formal rules, using the powerful Logical Model Processing (LMP) technology. Advanced multi format documentation generators are also included into the standard distribution of the tool.



5 Reduce the gap between System and Software engineering

Similarly, many discrepancies are often observed between the system engineering models and their corresponding software architecture. A solution to minimize this concern is to use a common language to ease the communication between these too critical phases of the life-cycle.

Stood supports the AADL (Architecture Analysis and Design Language), a new SAE standard for modeling system and software architectures. The aim is to increase the rate of reuse of common components between these two activities, and to facilitate software to hardware integration.

www.aadl.info