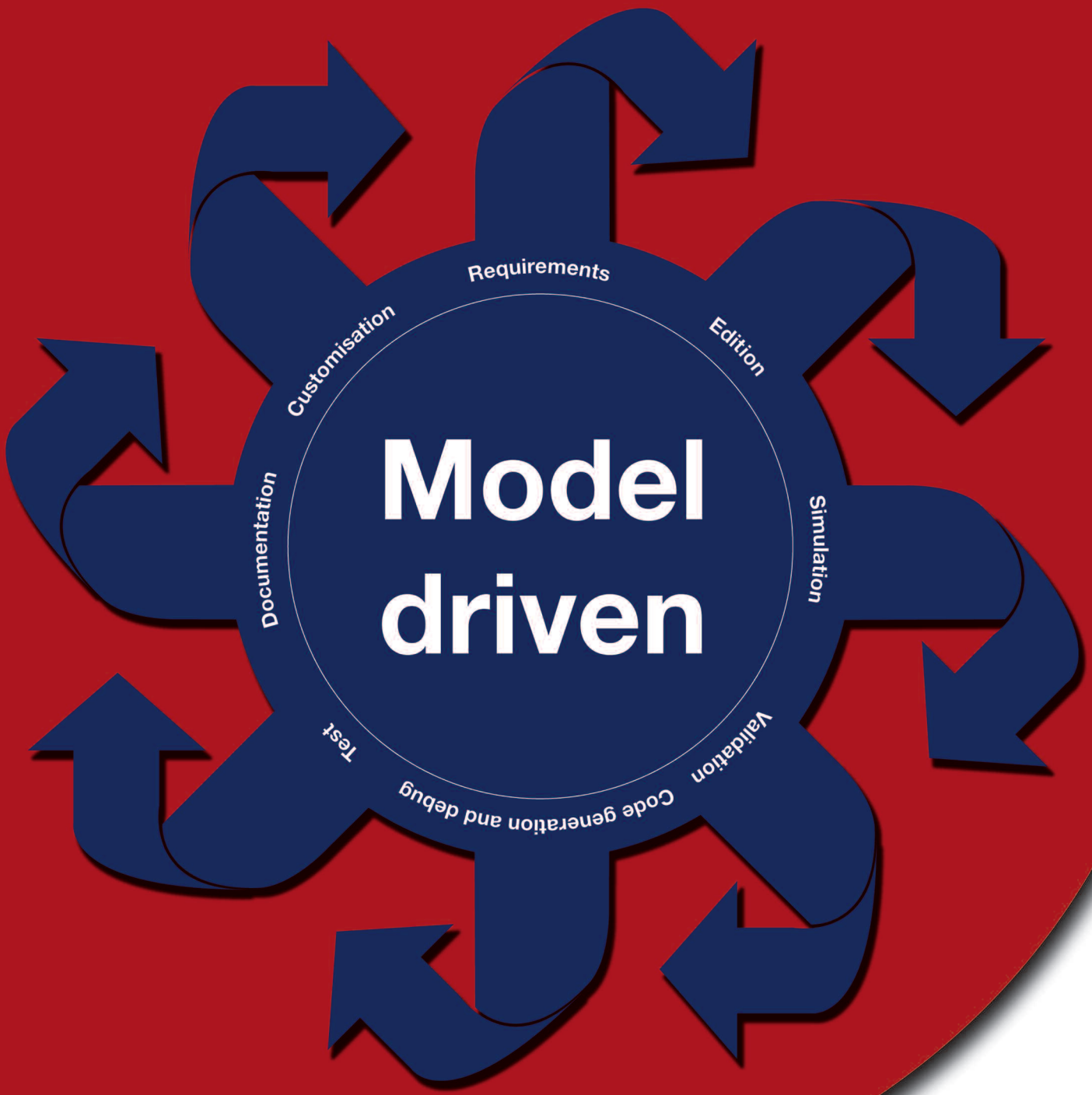
Model your inspiration

Embedded software modelling tool



PRAGMADEV

real time development tools

Model Driven Engineering Tool

Since the beginning of mankind great innovations have come from dreams. These dreams have first been illustrated with sometimes a high level of details such as the planes drawn by Leonardo da Vinci. This was basically a model of a plane that was implemented 400 years later, and that illustrates why we consider Model Driven Engineering to be a very natural approach for a development process.

A good model is abstract enough so that creativity is not slowed down by excessive details but precise enough so that it can be discussed and verified. This also illustrates there can be several levels of modeling: very informal to express an idea or a concept, semi-formal to make a link from the idea to the implementation, and formal to allow full verification before implementation.

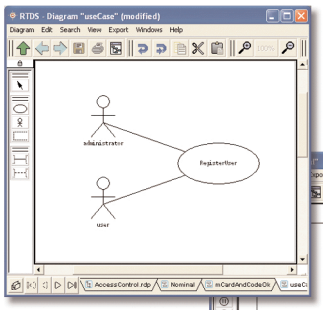
No matter how familiar you are with model driven development, Real Time Developer Studio will help with all approaches from a beginner with informal models, through intermediate with semi-formal models, and on to advanced expert with fully formal models.

Object orientation is not a universal solution for all types of development, which is why Real Time Developer Studio supports a functional approach as well as an object oriented approach.

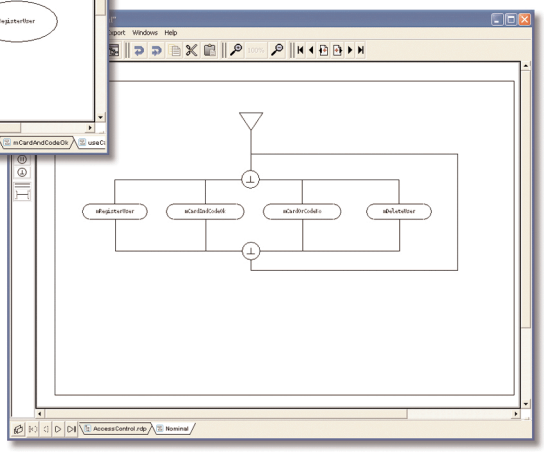
Real Time Developer Studio is based on recognized international standards and combines them to provide a consistent and flawless solution dedicated to real time and embedded systems.

Requirements

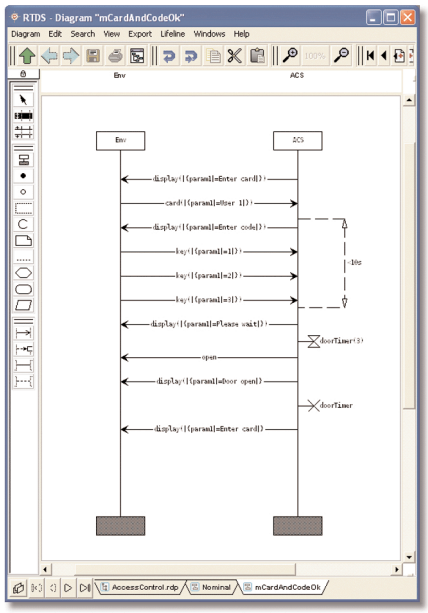
Requirements is the starting point of any project and is very often neglected because it is not formally expressed. **Model driven development** introduces high level diagrams such as Use cases and Sequence diagrams to help formalize the requirements written in natural language. These diagrams can be later re-used in the development process in order to check conformance to the requirements.



Use cases describe the actors involved



Build a scenario roadmap

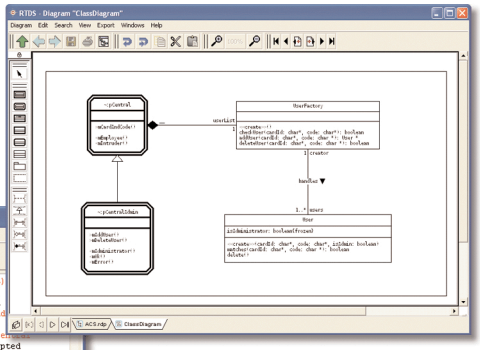


Further detail the scenarios

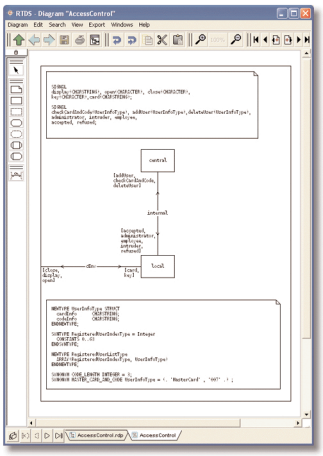
Because the needs are very likely to evolve during the development process, it is important to be able to track the impact of any modification to the requirements. **Real Time Developer Studio** offers a bridge with traceability tools to handle requirements management.

Edition

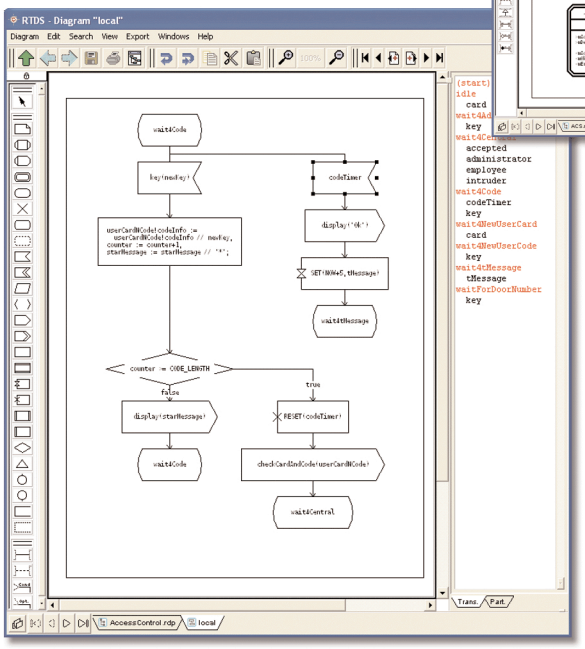
Easy to use graphical editors with contextual help and syntax verification help to build up a consistent system. An internal dictionary of already defined elements allows faster editing. A transition and a partition browser help to quickly navigate easily through the model. All diagrams are stored in separate XML files.
The tool includes a graphical diff and an automatic merge utility to be integrated with configuration management tools.



Object orientation view



Functional view



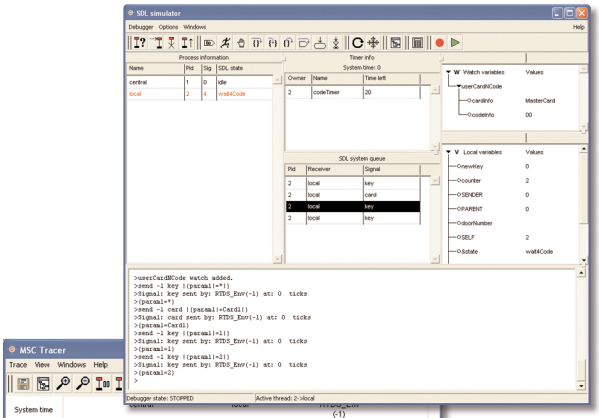
Advanced behaviour description

Editors are common to all types of modelling: informal, semi-formal, and formal.
The different diagrams allow developers to work on an object oriented or functional approach.

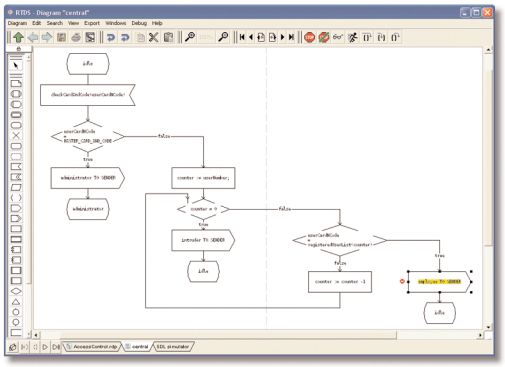
Simulation

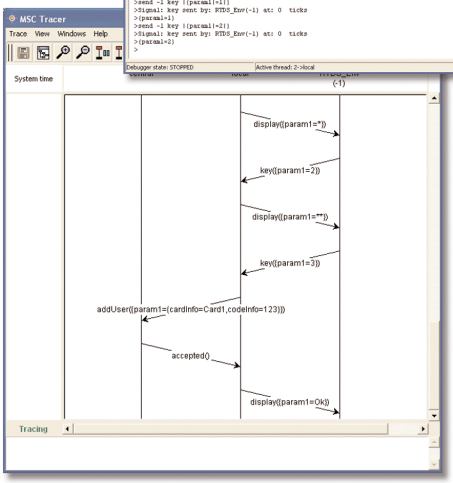
Formal and semi-formal models can be simulated to verify them. Semi-formal models simulation will use code generation to be executed on the host platform, and formal models will use the built-in simulation kernel. Semi-formal simulation is based on the selected target simulator semantic that can be a RTOS simulator or a processor simulator (ISS). Formal simulation offers full control over the execution of the system from discrete time execution to real time execution, changing states, changing variable values, generating signals, deleting signals and so on...

Model debugger



Step in the model



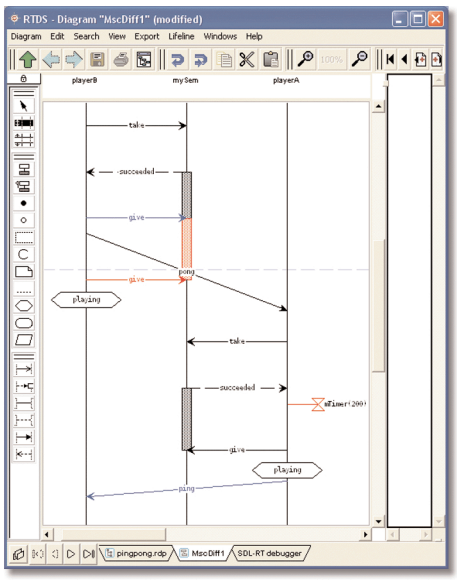


Model tracer

Validation

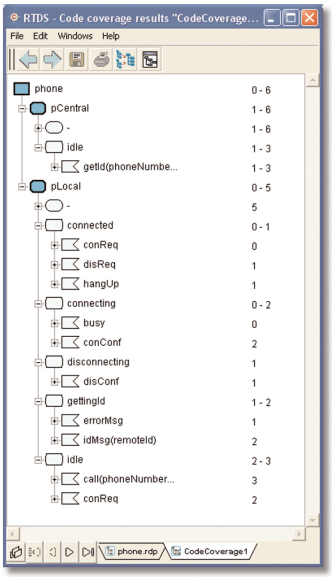
PragmaDev's Real Time Developer Studio offers several ways to validate a system:

- Against the requirements
  Because Real Time Developer Studio supports diagrams to express requirements, it is possible to validate conformance to the requirements.



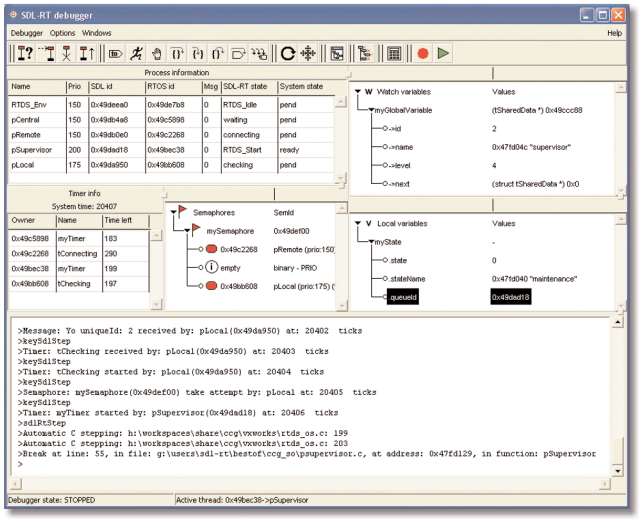
Compare model traces to check conformance and non-regression

Make sure all possible situations have been considered with the coverage viewer.



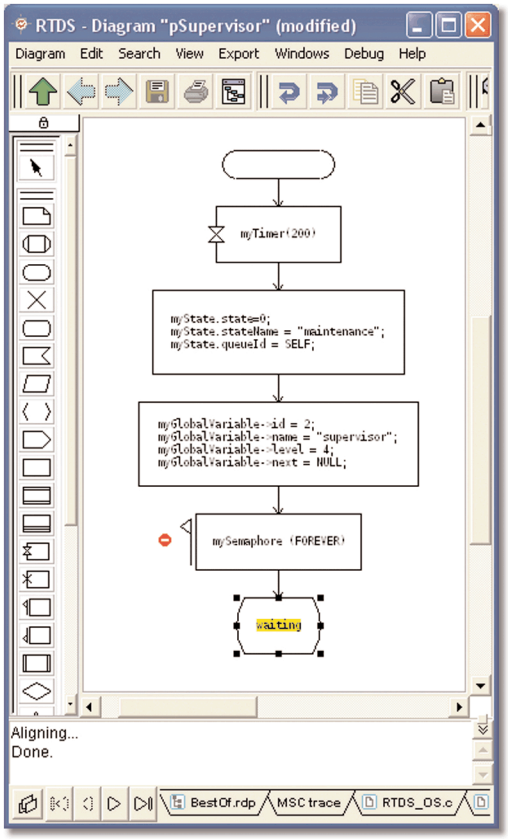
Code generation and debug

No matter how formal is the model, code can be generated. Code generation process can be customized to fit the target development environment. Integrations with the most popular operating systems is provided in the standard distribution. Graphical debugging at model level on host and target is also possible with the built-in debugger and cross-debugger integrations. Last but not least graphical traces are possible in debug mode as well as in release mode.

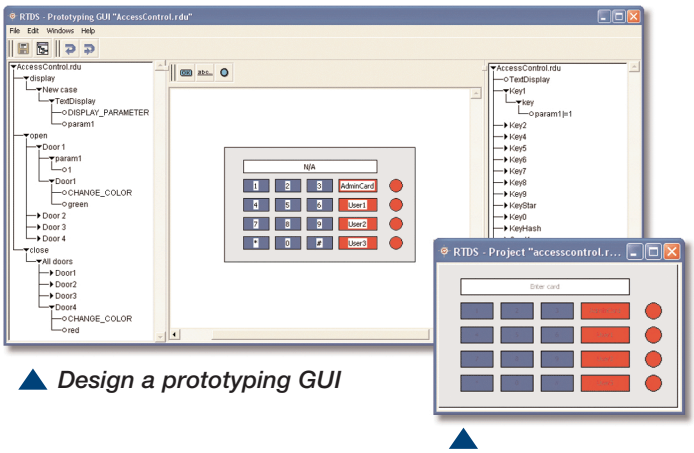


Debug the generated code

Debug on target in the model

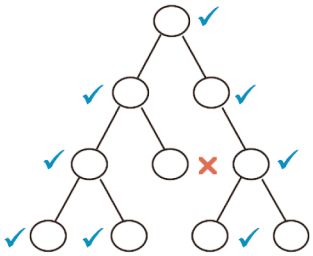


- With a prototype
  Real Time Developer Studio has a built-in prototyping GUI design tool in order to easily interact with the model. Executing the model with an end-user interface looking like the real system enables developers to verify that the model behaves as expected by your customer.

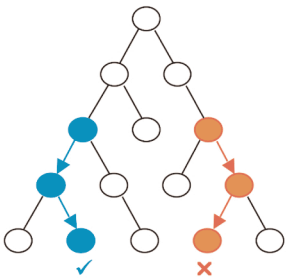


Design a prototyping GUI

It will automatically connect to the model



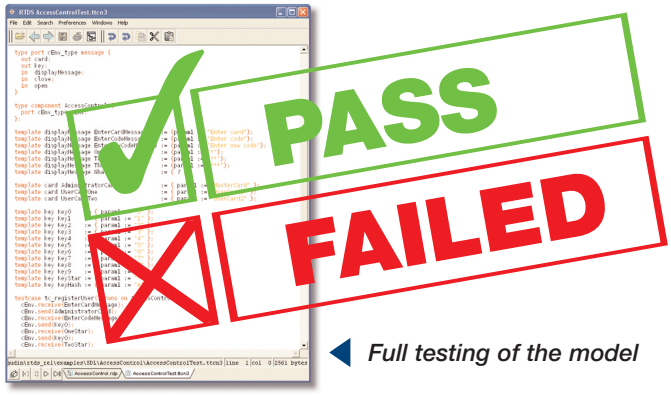
Static rules are verified in all possible scenarios



Dynamic rules are verified in all possible scenarios

- Against unexpected scenarios
  Real Time Developer Studio integrates exhaustive simulation and verification of formal models based on open recognized third party technologies. Observers describe static and dynamic rules to be verified in the system in all possible situations. Whenever a rule is considered violated or verified -depending on the rule objective- a graphical scenario can be generated for further analysis.

Test



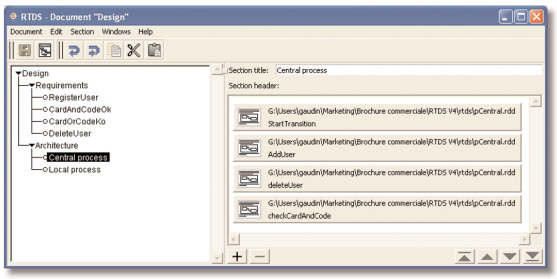
Full testing of the model

Latest release of RTDS introduces support of a standard testing language. Models can be simulated against test suites.

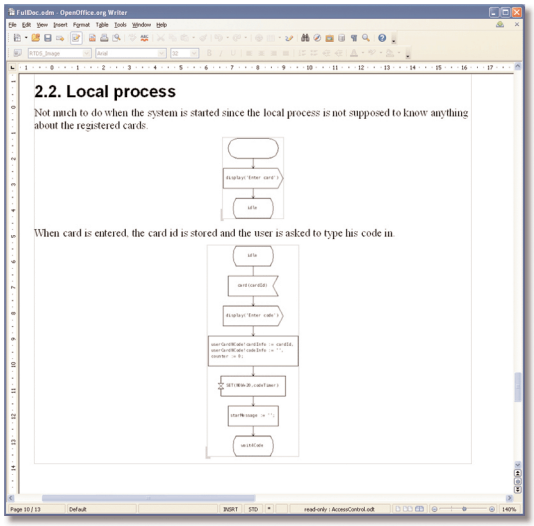
Traces and breakpoints can be set within the model or within the test suites and stepping is available on both sides because the bug might be in the model or in the test suite.

Documentation

Full documentation of the model can be generated automatically from the tool to RTF, Open Document, HTML, or SGML formats. Full or selected parts of the diagrams are automatically generated with their associated comments including index and table of contents entries.



Organize the documentation structure.

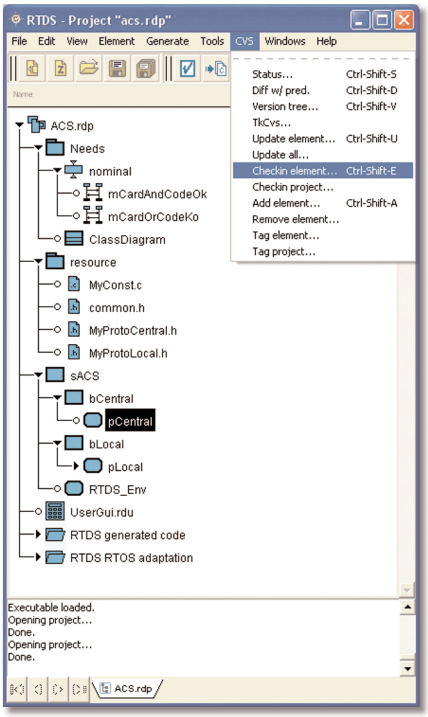


Documentation generated automatically.

The generated styles are customisable so that the generated document conforms to your corporate layout.

Customisation

Should you need a specific feature, PragmaDev can provide tool customisation to perfectly fit your development environment. The newly introduced features will be incorporated into the standard product so that they are maintained in future releases. Because customisation to add a feature or modify the way that the standard tool operates is often quick and inexpensive, it must be seriously considered as the productivity improvement can save a lot of time over the course of a project.



Customize and adapt to your environment.

Openness

Real Time Developer Studio is an open tool on several aspects:

- It can import and export the models in several formats.
- Each element of the project is stored as an **XML** file.
- *RTDS* features can be called from a command line.
- *RTDS* features can be called from a CORBA broker.

Licensing and supported platforms

- Real Time Developer Studio runs on Windows, Solaris, and Linux platforms.
- Real Time Developer Studio uses floating licenses that can be used anywhere on a local area network.
- Generated code and templates are delivered as source code with no royalties.

Technologies

- ASN.1: Abstract Syntax Notation One is standardized by International Telecommunication Union under reference X.680.
- IF: Intermediate Format is defined by Verimag Labs.
- MDA: Model Driven Architecture is standardized by Object Management Group.
- SDL: Specification and Description Language is standardized by International Telecommunication Union under reference Z.100.
- SDL-RT: Specification and Description Language Real Time combines UML, SDL and C/C++ languages within one consistent model.
- UML: Unified Modeling Language is standardized by Object Management Group.
- TTCN-3: Testing and Test Control Notation is standardized by the International Telecommunication Union under reference Z.140.

PRAGMADEV
real time development tools

PragmaDev
18, rue des Tournelles
75004 Paris France
Tel : +33 1 42 74 15 38
Fax : +33 1 42 74 15 58
www.pragmadev.com

sdl SDL RT UNIFIED MODELING LANGUAGE UML