TELECOM • AVIONIC • SPACE • AUTOMOTIVE • SEMICONDUCTOR • IOT • MEDICAL

PRAGMADEV SPECIFIER

PRAGMADEV DEVELOPER

PRAGMADEV TESTER

FOUR INDEPENDENT TOOLS TO MANAGE COMPLEXITY INHERENT TO DEVELOPING STATE OF THE ART SYSTEMS.

PRAGMADEV TRACER

PRAGMADEV STUDIO

PRAGMADEV STUDIO

PragmaDev Studio is a set of connected tools that helps Specifiers, Developers, and Testers to manage complexity in the development of today's systems.

A reduced time to market, quality products, with the best connectivity are the challenges development teams have to deal with. In order to be efficient all phases of the development tend to occur at the same time in an iterative process. PragmaDev Studio automates the whole development process and keeps the focus on high level requirements.

PragmaDev Studio will lead to:

- Early verification
- Early automated testing
- Improved quality
- Precise documentation
- Optimized performance
- Safer deployment
- Property verification
- Reduced time to market

PragmaDev Studio is our premium package that includes all other modules: PragmaDev Specifier, PragmaDev Developer, PragmaDev Tester, and PragmaDev Tracer. PragmaDev Studio is the ideal solution for all development phases from specification to validation.

In addition PragmaDev Studio includes a number of advanced features linking one module to another in a consistent environment.

MODEL BASED TESTING

Once the stakeholders have validated the system functionalities with PragmaDev Specifier, test cases can be generated automatically based on:

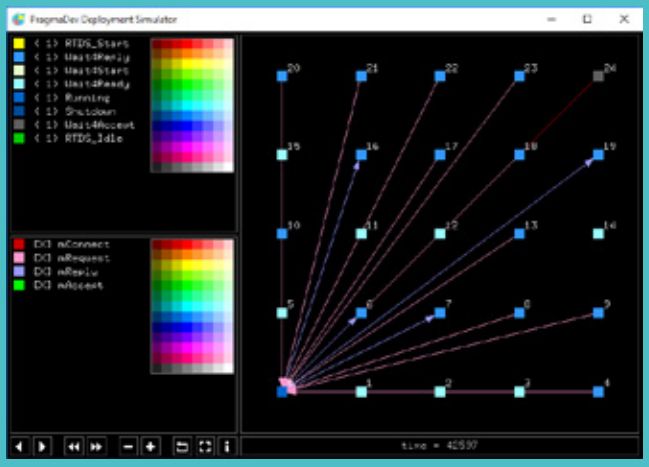
- Functional scenario
- Full coverage
- Single transition coverage
- Property verification

MODEL CHECKING

Thanks to the execution semantics of the modeling language, analysis of the model can be automated to verify properties on the model with third party technologies. PragmaDev Studio can export models to tools from well known research centers such as Verimag, LAAS, or CEA List.

DEPLOYMENT SIMULATOR

Mobile communication, M2M, and the Internet of Things deploy thousands or millions instances of small systems to build up a system of systems. PragmaDev Deployment simulator aims at verifying such a topology works correctly.



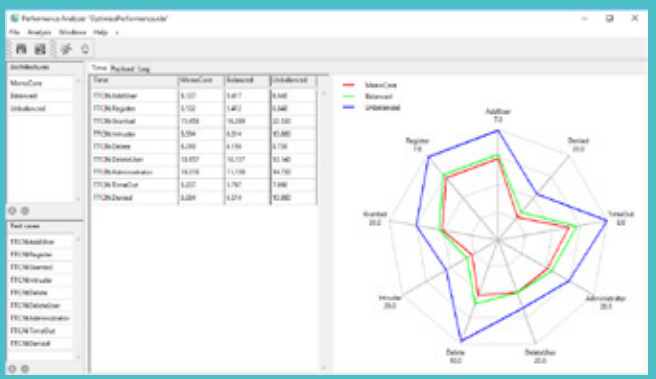
Simulate the deployment of numerous instances

REQUIREMENTS TRACEABILITY

PragmaDev Studio can import textual requirements and keep traceability from a requirement, to a part of the model, to test cases. Therefore the impact analysis is straightforward and it is possible to know which test cases should be run if a requirement is modified.

PERFORMANCE ANALYZER

Performance Analyzer will automatically find the best architecture of your system for timing or energy performance.



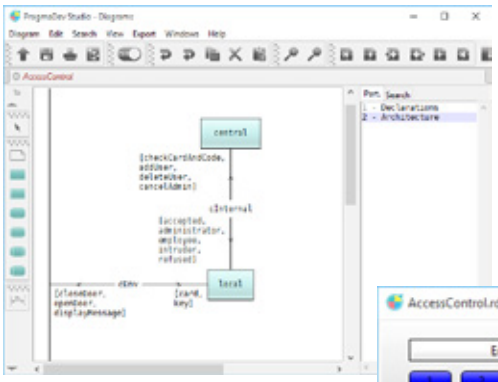
Identify the best architecture to optimize timing or energy performance

PRAGMADEV SPECIFIER

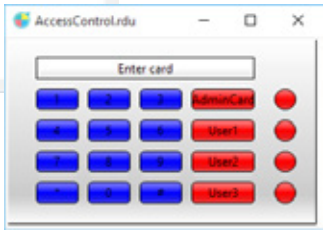
PragmaDev Specifier helps system engineers, architects, and specifiers to express their needs in a graphical and executable functional model.

A majority of the systems developed are not what was expected because of a wrong understanding of the requirements. PragmaDev Specifier aims at helping communication between the stakeholders early in the development process. It builds a bridge between the domains of expertise and clears out any possible misunderstanding.

PragmaDev specifier helps:
- **to communicate,**
- **to define a functional architecture,**
- **to verify functionalities dynamically,**
- **to make sure requirements are all covered.**



Graphical functional architecture

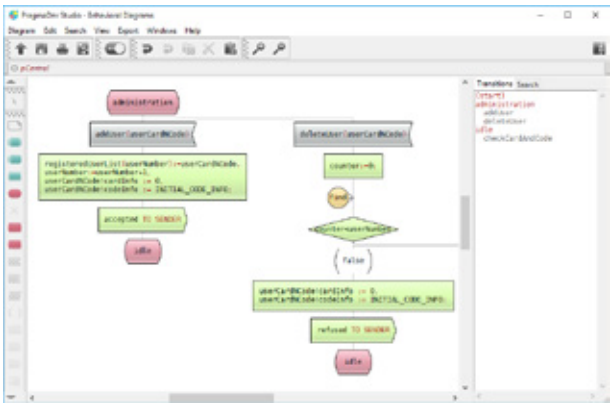


Create an interface to interact with the system

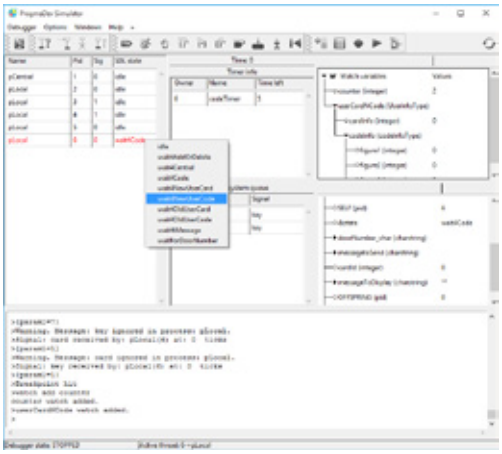
The model expresses several aspects of a system: the expected architecture and the behavior of the different elements. Because the model is executable, the powerful builtin simulator combined with the graphical user interface prototyping tool allows a non specialist to validate the behavior of the model. The resulting set of scenarios can be used as dynamic requirements or turned into validation test cases to be run on the implementation.

The resulting model is a reference for all the stakeholders. It can be used for prototyping, implementation, generating test cases, generating documentation.

Tomorrow systems will connect an important number of devices creating a substantial level of complexity. It is of the utmost importance to functionally clearly define what each subsystem is supposed to do from a static and a dynamic point of view. Ambiguity is not an option even for non safety critical systems.



Graphical description of the behavior



Model simulator to analyze the system execution

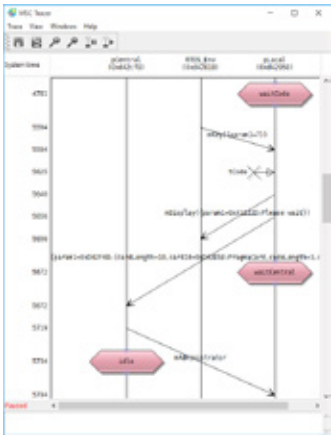
PRAGMADEV DEVELOPER

PragmaDev Developer helps software developers to define their software architecture and the concurrent behavior of the different agents.

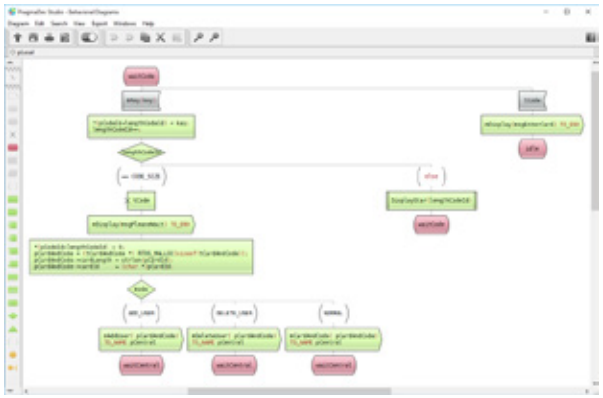
Because of the inherent complexity of development, and because the requirements are often changing, it is important to keep a good level of communication and understanding of the technical choices. Graphical modeling helps developers to interact with the previous phases of the development to handle iterations. Support for traceability information in the model keeps the link back to the requirements. With its automated testing capabilities, PragmaDev Developer is the perfect tool for continuous integration.

PragmaDev Developer raises the developers expertise by:
- **focusing on functional concerns,**
- **making a clear architecture,**
- **ensuring maintenance of the software,**
- **generating documentation automatically.**

Graphical symbols are used to describe the architecture and the high level behavior of the different agents. Low level behavior is written in C or C++ in the graphical model so that the code and the model are always fully synchronized. Once the model is validated on host the code can be generated to run on target. The generated code can run by itself with the provided scheduler, or on any RTOS.



Execution traces between agents in the system



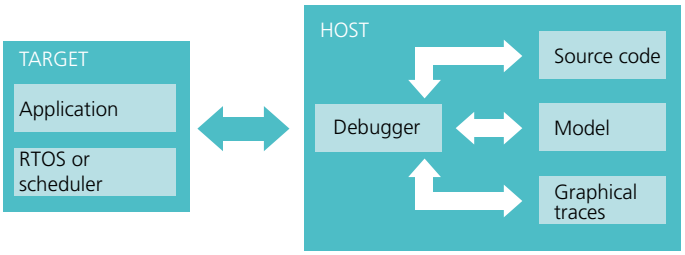
Graphical behavior embedding C/C++ code

The generated code can be customized and we have taken special care to keep it legible and understandable.

The modeling technology is ready to use and the engineering team can focus on their work instead of their tooling. The underlying concepts to build the model are the ones engineers are familiar with: state machines, message queues, timers, semaphores, procedures...

Legacy is easy to integrate because C/C++ code is part of the modeling technology.

Connection with debuggers and cross-debuggers allows model debugging on host and target.



Generic model debugging architecture

The code generator can embed:
- tracing information to connect to PragmaDev Tracer,
- back trace information to analyze the last past events,
- coverage information to make sure all branches have been tested.

PRAGMADEV TESTER

PragmaDev Tester helps testers to design and execute their test cases. PragmaDev Tester is ideal for functional black box validation testing, but also covers intermediate white box testing phases such as integration testing, or unit testing.

Because requirements are changing rapidly, it is of the utmost importance to test automatically the specification as well as the implementation. PragmaDev Tester helps to write abstract test cases that can be run against a high level specification as well as an implementation.

PragmaDev Tester enables:
- **testing a model as well as an implementation,**
- **testing non regression automatically,**
- **ensuring conformity.**

PragmaDev is based on TTCN-3 standardized language dedicated to test. Any event driven system can be tested with PragmaDev Tester, and the test cases are tester independant. In addition to its textual notation PragmaDev Tester can generate interaction diagrams from the test cases for documentation.

Test cases are:
- Portable
- Executable
- Automated

Test cases can be written manually or generated automatically from a set of scenarios or directly from an executable model (Model Based Testing feature in PragmaDev Studio).
Ready to use TTCN-3 test cases can be downloaded from standardization bodies such as ETSI to check conformance to a standard.

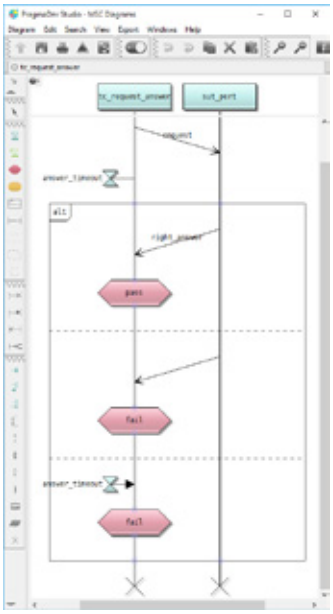
TTCN-3 splits the test cases in several parts:
- Declarations with all the necessary built in or ASN.1 data types,
- Templates to easily match what is sent and received,
- Test cases to describe the set of sends and receives,
- Control part to automatically execute a selected set of test cases.

The tool consists of a powerful TTCN-3 editor with graphical representation of the test case, and a C++ code generator to execute the test case on a tester. Test execution can be debugged during execution because of the integration with debuggers.

It is possible:
- to set breakpoints,
- to view internals,
- to trace graphically.

PragmaDev Studio has a TTCN-3 simulator that can run against an executable model designed with PragmaDev Specifier or PragmaDev Developer.

A simple test case



Graphical view of a test case

PRAGMADEV TRACER

PragmaDev Tracer helps to express requirements graphically, define properties, and trace execution.

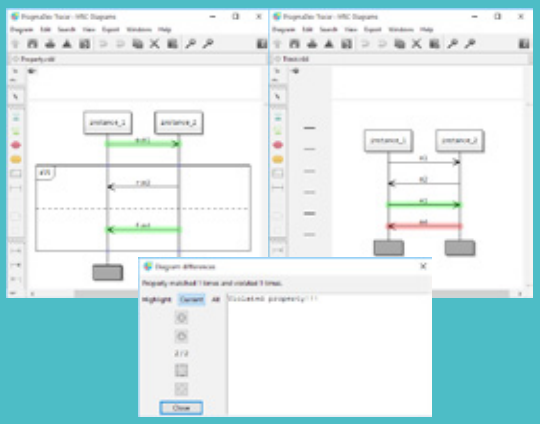
FREE MODULE

M2M communication and the Internet of things bring systems to interact more and more. Even though each module can be quite simple, the complexity comes because of the number of instances put together. PragmaDev Tracer helps to visually deal with these interactions between entities.

PragmaDev Tracer helps engineers to:
- Express the requirements of your system,
- Trace your system behavior online or offline,
- Verify the trace matches the requirements.

PragmaDev Tracer is freely available and fully integrated with all the modules within PragmaDev Studio.
It features:
- Easy edition of the expected behavior of your system,
- Definition of the functional and non-functional properties of your system,
- Tracing online via a socket or offline in a file,
- Matching the traces against the requirements and/or the properties.

Graphical representations for:
- Tasks,
- States,
- Messages,
- Semaphores,
- Timers,
- Objects,
- Alternatives,
- Loops,
- Time constraints,
- Chain constraints.



A property violation detected on an execution trace

Three levels of verification with filtering capabilities:
- Diagram comparison.
- Match requirements on traces with the support of relative time constraints, and inline expressions such as alternatives and loops.
- Match properties on traces using the Property Specification Chart notation.

Integration in your testing or development environment
The tracer can be started in text mode and in batch mode and control commands can be sent through a socket. C macros are also available to easily generate traces in a file or in memory.

Property definition
Within the PSC language, a property is seen as a relation on a set of exchanged system messages, with zero or more constraints. PSC may be used to describe both positive scenarios (i.e., the "desired" ones) and negative scenarios (i.e., the "unwanted" ones) for specifying interactions among the components of a system. PSC has both formal notation and operational semantics.

CORPORATE VALUES

VISION

The number of connected objects in the near future will reach a new level. Because of the inherent complexity due to the dynamic behavior of communicating devices, and the cost of deployment of a real configuration, modeling will become a must have. Make sure the requirements are well understood to ensure interoperability. Make sure the system will be maintainable including all its possible variants all along its lifetime to ensure backward compatibility and functional stability. Automatically explore execution paths generated by the complexity of event driven systems. To deal with these new challenges precise and specialized modeling technologies will be used.

MISSION

Our mission is to help our customers to manage complexity in their development. This includes tooling, training, coaching, and customization.

BENEFITS

PragmaDev toolset addresses three main issues:

- **Complexity**

The increase in processing power and the necessary connectivity of all new systems generate a high complexity level. Models help managing that complexity.

- **Maintainability**

The cost for the development of the first version of a system is a small part of the total cost of ownership. But the technologies used to develop the first version will have a huge impact on future spendings. Our modeling and automated testing technologies ease evolution, and regression testing is made effortless.

- **Communication**

Whether your client is internal or external, developing a system is basically translating a requirement into an implementation. The technical cultural background differences sometimes make communication very harsh and inefficient. Modeling is a means of communication that can be understood by all, a bridge between what to do and how to do it, the link between the teams.

CUSTOMIZATION

Our customers are our partners because we believe we will be successful if our customers are successful. For that, especially during the first project, we always work hand in hand with our users. This usually leads to the development of new features that will help integrate our tools in other technical environments.

PROFESSIONAL SERVICES

The first use of a new technology is always a challenge. No matter how good it is, if misused, it might lead to a waste of time and money. To reduce the startup time and make sure the technology is efficiently used, we provide highly skilled consultants.

SECURE YOUR INVESTMENTS

- PragmaDev Studio source code and documentation are deposited at the Agency of Protection of Programs.
- PragmaDev offers flexible licensing mechanisms. It is possible to buy a license or to pay a monthly subscription.
- All the technologies used in PragmaDev Studio modules are recognized international standards allowing import and export with other tools.

PRAGMADEV
modeling and testing tools

18, rue des Tournelles 75004 Paris France
Tel : +33 1 42 74 15 38
http://www.pragmadev.com
mail : info@pragmadev.com