CITRUS

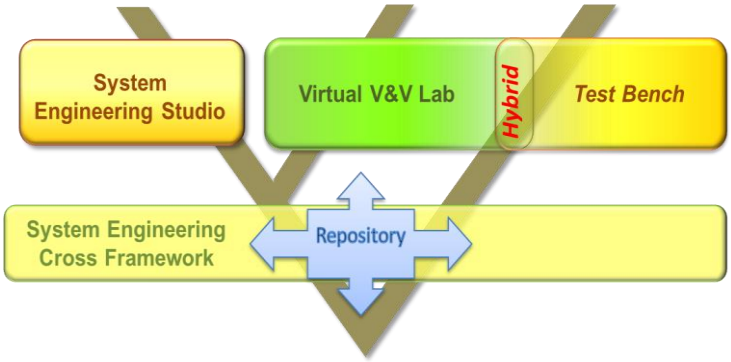
A software suite to develop complex systems with zest!

Citrus is a product that fully supports the development cycle of a system, with an efficient and innovative approach based on modeling and simulation.

At each stage of the development, from the early engineering studies to the full-system test bench, Citrus allows to **verify and validate** the system of interest through simulation.

Citrus offers an **open and modular** framework with a set of components assembled in order to provide consistent, high-performance and easy-to-use solutions for each phase of the development cycle:

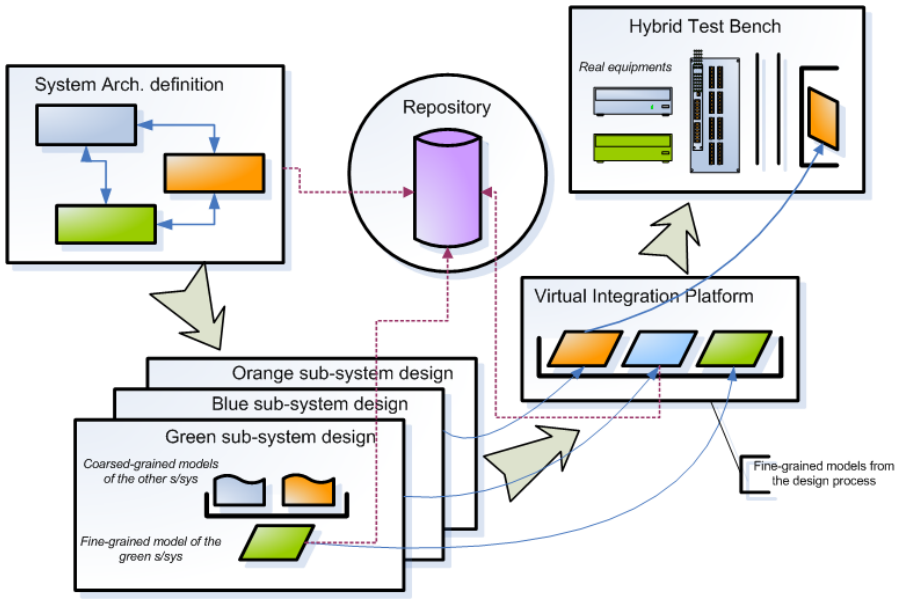
- **System Engineering Studio** captures the system architecture, its interfaces and its breakdown structure into sub-systems/equipments.
- **Virtual platform** integrates the whole system increment by increment and perform verification and validation activities.
- **Hybrid Test Bench** infrastructure allows to easily build the real system test bench, in interaction with a simulation of the system environment.



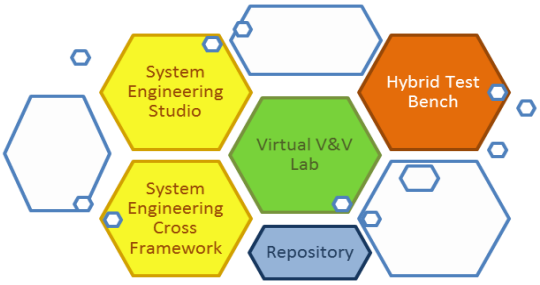
An integrated approach

Citrus ensures **data continuity** throughout all the steps of the development cycle. Each work performed by a team becomes available through a shared **repository** to other teams without any adaptation. This allows to:

- Establish a short loop between the different parties and drastically reduce the cost and time spent on corrections.
- Ensure integrity, consistency and comprehensiveness of data in order to avoid information loss through the whole process.
- Master the changes brought at each delivery stage. The results of each phase are managed in configuration with traceability so that the impact analysis may be automated.



Modular and user-oriented



Each solution is completely autonomous and can be used as a unit to take advantage of the efficiency and usability of the tools. Solutions can be integrated gradually into the full suite which then guarantees optimum productivity.

To focus on the users needs, Citrus introduces the notion of profiles that specializes the tool for a particular business domain. The suite is then operated according to a point of view corresponding to concepts commonly managed by the users.

For instance, with the Integrated Modular Avionic (**Arinc 653**) profile, it is possible to directly manipulate concepts such as partitions, ports or the real time characteristics of the platform and its applications.

System Repository

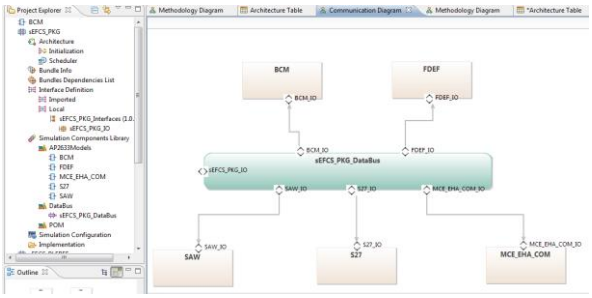
The **System Repository** is a key component of the **reuse strategy**. Each element (document, interface, model, source code, etc.) created and managed by the different solutions is manipulated as a versioned bundle. An assembly of bundles is also managed as a bundle; reuse could thus range from the detailed definition of a specific interface to a complete simulation application.

The System Repository also provides:

- Configuration and variants management.
- Traceability between bundles and versus requirements.
- Dependency management.

System Engineering Studio

Using the engineering studio, the designers can build the system breakdown structure either with a top-down approach by defining the elements of the system, or with a bottom-up approach by assembling elements that already exist in the Repository.



The system is described by the mean of a model which uses the natural concepts of the system engineer. With a set of user friendly specific editors, System Engineering Studio provides the power of **Model Driven Engineering** (consistency checks, automatic transformation …) without confusing users with complex and tedious displays.

System Engineering Studio allows to separately describe the **functions** of the system and the **platform** on which they are executed.

Some of the services offered:

- Editing and browsing the system composition (zoom, point of view…).
- Definition of the interfaces with physical value and/or formatted value (CAN, A429, AFDX …).
- Configuration and deployment on the platform (allocation of resources, consistency checks, etc.).
- Configuration and deployment of a simulation test bed for the system under interest.
- Code generation for adapters between elements.
- Documentation generation.
- Export/import to/from SysML & AADL.
- Analysis of the differences between versions and the associated impacts on an upgraded assembly.

An open framework

Citrus is designed as an open and modular framework that can easily be connected with other external products using different formats (model or data).

Citrus provides a native interface with the major market-standards (such as Simulink©). For the other ones, a specific interface can be added via the development of a plug-in extension.

This aims at avoiding any vendor-locking situation. If an efficient component already answering to a specific need can be found on the market, the objective will be to encapsulate and integrate it on the Citrus suite.

- Static analysis on the deployment (resources usage…).

Virtual V&V Lab

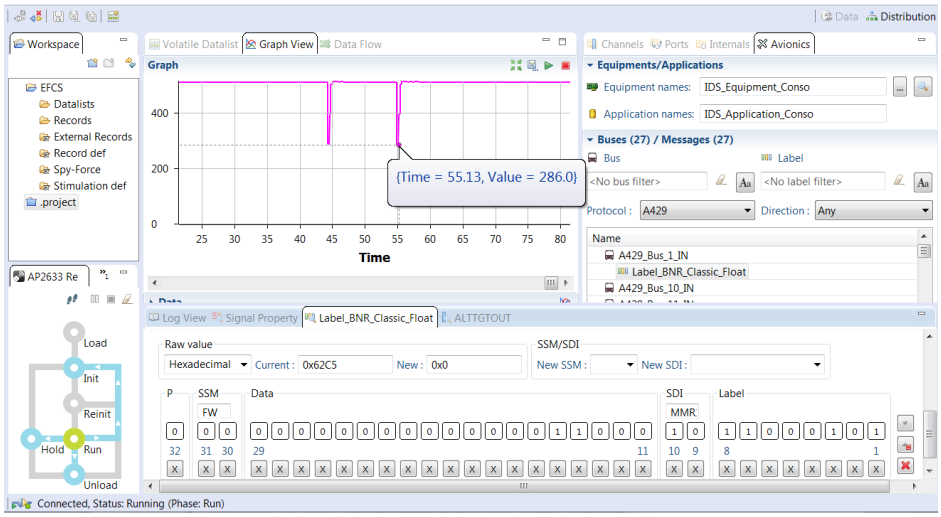
The Virtual V&V Lab is an environment that manages the models of the equipment under design and provides a simulated environment for its **verification and validation** activities.

In the early phases of the development cycle, the whole assembly is executed in a simulation in order to validate and mature the equipment detailed specifications, in an integrated environment with functional loops between the equipment outputs and inputs.

Later in the development cycle, Virtual V&V Lab allows building a highly representative simulation of the system. Integration tests can then be designed and executed on this platform to check the interfaces conformity and the dynamic of the system: these same tests can be conducted on the real physical bench while most of the issues have already been corrected, allowing to **drastically reduce the time and effort on the bench**.

The simulated environment of the equipment under design can be specifically built or automatically generated by the System Engineering Studio from the system description developed by the designer and available in the Repository.

For performances reasons (data volume / computing time) or due to the models/applications heterogeneity, the simulation can be distributed on an Ethernet-based network of computers.



Some of the services offered:

- Models generation from the commonly used modeling tools of the industry: Simulink©, Scade© …
- Environment for model edition (C, C++, FORTRAN, Ada, Java …).
- Dynamic loading of models: no compiling activity of the simulation is required; a model can be updated by a simple copy.
- Support for the standard bus formats: CAN, A429, AFDX, A1553 …
- Monitoring and control of the simulation (visibility on all parameters, raw or formatted, with public or internal interfaces, time jump, activation/deactivation of models and data exchanges …) thanks to a powerful and user-friendly HMI.
- WYSIWYG synoptic editor to display the running simulation and interact with it.
- Parameters recording supporting various Quality of Service
- Simulation Instrumentation: real-time modification of parameter values via formulas or recorded datasets…
- Design and execution of tests scenarios.
- Co-simulation between the model under design executed in the original modeling tool (Simulink©, Scade©, Modellica©, Scilab© …) and the simulated

Performances

*Citrus is used in the aeronautic field to build significant simulations composed of more than 150 models that are scheduled with time periods between 50 usec and 50 ms and dealing with a number of signals varying from 500 000 up to **1 million**.*

The common deployment platform for this kind of simulation consists in 4 up to 20 computers.

environment.

- Smart wrappers to connect data that are a priori not compatible (different units, formatted versus unformatted data, different types…).
- Distribution management: deployment, supervision of the state of calculation nodes…
- Management of real-time synchronization by group of applications.
- Fault- tolerance and failover: detection of the loss of a simulation federate and automatic re-launch.
- Non-intrusive supervision of the real-time execution of models.

Hybrid Modular Bench

This solution consists in a new generation bench that leans on:

- A highly modular architecture where each function is supplied as a plug-in: driver, bus codec, protocol, test language … As a result each new function can be integrated without changing the existing.
- A native integration with the virtual integration platform which allows to mix hardware and simulation on an hybrid platform.
- A distributed and scalable architecture that deals with heterogeneous equipments. The bench is made of calculation nodes on an Ethernet network.
- An hardware abstraction layer that allow to easily manage I/O boards. At the application level, engineers can manage complex data structures without any knowledge of the hardware processing these data.

Some of the services offered:

- Deployment of new functionalities (card, equipment…) without stopping the bench and loosing the test workload achieved at this stage in the session.
- Start of the bench with an empty configuration: the bench can then be used very early in the development cycle and the elements are gradually added.
- Live connection or instrumentation of data: an input/output of a card can be connected to a variable of the simulation while it is already running.
- Use of different scripting means (Python, Lua, C++ …).
- Environment to edit simulation models.
- Edition and integration of synoptic displays connected to the variables of the bench (Adobe AIR/Flash MMI).
- Graphical user environment to configure, visualize and exploit the bench. The solution enables the bench operations through Internet firewalls (concept of extended enterprise).
- Multi-OS platform support.