Reference Card

SCADE Products Glossary

SCADE® 2025 Products Family

SCADE Products Glossary Overview

This glossary defines terms related to SCADE products family. The terms may have other meanings in different contexts and may apply only to SCADE Suite (tag: **SUITE**), SCADE Display (tag: **DISPLAY**), SCADE Architect (tag: **ARCHITECT**), SCADE LifeCycle (tag: **LIFECYCLE**), SCADE Solutions for ARINC 661 (tag: **ARINC661**), SCADE Test (**TEST**), SCADE Rapid Prototyper (tag: **RAPIDPROTO**), or several (tag: **SUITE**/**DISPLAY**). Terms are also classified within the following categories: Software Design, Verification, Generation, System Design, Management, and Certification.

A

A429
`Design - ARCHITECT`

See ARINC 429.

A653
`Design - ARCHITECT`

See ARINC 653.

Absolute Position
`Management - DISPLAY/RAPIDPROTO`

The position, in user unit or in pixels, of an object in the layer, defined by its absolute X position and its absolute Y position.

Activation
`Design - SUITE`

A feature that allows executing an operator when the activation condition is true.

Active state
`Design - SUITE`

See SCADE State Machine.

Activity
`Design - DISPLAY/RAPIDPROTO`

A Boolean variable that defines if the current clipping mask is active or not.

Adaptor
`Generation - SUITE`

See RTOS Adaptor.

Advance
`Design - DISPLAY/RAPIDPROTO`

The distance between the origin (or start) of a glyph and the origin of the next glyph. Called vertical advance when the text is written vertically, like in Chinese, Japanese or Korean (CJK) scripts.

AFDX
`Design - ARCHITECT`

The acronym of Avionics Full Duplex Switched Ethernet. This data network is a specific implementation of ARINC Specification 664 Part 7, defining a reliable real-time environment for the use of Ethernet networks for the communication of avionics data.

Algorithm design
`System - SUITE`

A type of design developed in an appropriate notation (*e.g.*, in Scade or in Simulink®) to create a mathematical process in solving problems.

Alpha-blending
`Design - DISPLAY/RAPIDPROTO`

An effect applied on the pixel color to mix the defined pixel color with the background color, thus allowing to modify the levels of transparency and opacity of pixels.

Animation

`Verification - DISPLAY`

A mode that allows animating the specification quickly to verify its design. The specification can also be animated on a “functional” mode if plugs are defined in it (the image of the specification changes according to the values of the plug variables).

Animation scenario

`Verification - DISPLAY`

A scenario applied to a variable in order to animate it.

Annotation

`Design - SUITE`

A type of comment that can be associated with any SCADE Suite entity (package, types, constants, operators, inputs, outputs, etc.) for traceability needs. As for comments, annotations can be propagated in the generated code.

Annotation file (ANN)

`Management - SUITE`

A file that is associated with an XSCADE file and that contains annotations on elements defined in the XSCADE file.

Annotation type file (ATY)

`Management - SUITE`

A file that defines the types of annotations that can be created when editing a model.

Anti-aliasing

`Design - DISPLAY/RAPIDPROTO`

A technique that enables to smooth the contour of an object by mixing the object color with the background color by applying Alpha blending. The color interpolation is performed by applying a filter computing the color opacity (Alpha value) from the position of the pixel toward the core of the line.

Architecture design

`System - SUITE`

A type of design of which objective is to lay the foundations for the development of the Scade Low Level Requirements (LLR). A good Scade architecture is composed of data type definitions, top-level operators, and their connections, which ensure stability, maintainability, readability and efficiency.

ARINC 429

`Design - ARCHITECT`

A data bus standard for aircraft avionics, that defines both the physical and electrical interfaces of a two-wire data bus, and a data protocol to support an aircraft's avionics network.

ARINC 653

`Design - ARCHITECT`

A software standard for space and time partitioning in safety-critical avionics real-time operating systems (RTOS). It describes a programming and configuration interface that facilitates the independence of avionics applications with regard to hardware. Each application software is called a partition.

ARINC 661

`Design - DISPLAY`

A standard that defines interactive interfaces to Cockpit Display Systems (CDS) used in modern aircraft installations.

ARINC 664-P7

`Design - ARCHITECT`

See AFDX.

ARP 4754

`Certification - SUITE`

A standard that discusses the certification aspects of highly integrated or complex systems installed on aircraft, taking into account the overall aircraft-operating environment and functions. It addresses the total life cycle for systems that implement aircraft-level functions. It excludes specific coverage of detailed systems, software and hardware design processes beyond those of significance in establishing the safety of the implemented system.

Array

`Design - SUITE`

A one-dimensional sequence of values sharing the same type. An array of [n] elements is defined statically and always starts with the 0 (zero) index. Matrices or other higher dimension arrays can be modeled with array nesting (array of array of …).

ASAM MCD-2 MC

`Generation - SUITE`

ASAM MCD-2 MC (formerly known as ASAP2) is a data definition standard proposed by the Association for Standardization of Automation and Measuring Systems (ASAM). ASAM MCD 2 MC is a standard description you use for data measurement, calibration, and diagnostic systems. SCADE Suite enables the export of an ASAM MCD-2 MC file containing information about SCADE Suite models during the code generation process.

Assertion

`Verification - SUITE`

Assertions introduce environment information in SCADE Suite models and accelerate formal verification by reducing the input variation domains. It is possible to declare *Assume* assertions to observe inputs and the past of outputs or *Guarantee* assertions to observe inputs and outputs.

Assume

`Design - SUITE`

One of both observers that characterize a contract. The assume expresses the contract's pre-condition and observes the inputs and the past of the outputs.

Asynchronous mode

`Verification - SUITE`

An integration mode for co-simulation between behavioral logic and graphical panel components. In this mode, the graphical panel runs independently from behavioral logic model (*e.g.* when simulation is paused, graphical panel still reacts to user interaction). See Synchronous mode.

B

Background

`Design - DISPLAY/RAPIDPROTO`

The part of a model that can display a bitmap image and a checkerboard. As layers have a transparent background, the specification background is always visible.

Baseline

`Design - DISPLAY/RAPIDPROTO`

An imaginary line upon which each character rests. Characters that appear next to each other are (usually) lined up so that their baselines are on the same level. Some characters extend below the baseline (``g'' and ``j'', for example) but most rest on it.

BC

`Verification - TEST`

Acronym for Branch Coverage.

Bezier line

`Design - SUITE`

A parametric curve used to represent transitions in state machines.

Bi-font

`Design - DISPLAY/RAPIDPROTO`

A text-based primitive that allows displaying formatted numbers and associating two different fonts (*e.g.*, one of the integral part and one for the decimal part).

Bitmap

`Design - DISPLAY/RAPIDPROTO`

A rectangular array defining the color to apply for each pixel.

Black-box simulation

`Verification - SUITE/DISPLAY`

Simulation of (a piece of) a SCADE Suite model or SCADE Display specification in batch mode, without reference to its internal behavior. The goal is to verify how well the model/specification conforms to its requirements. Contrast is White-box simulation.

Block diagram

`Design - SUITE`

A structured notation consisting in a network of interconnected computing units called blocks. Each unit offers connection to receive inputs and connections to provide outputs. Graphical representation of block diagrams usually represent blocks as rectangle boxes with input pins representing the input connections and output pins to represent the output connections. In SCADE Suite, data-flow programs are graphically represented using block diagrams where blocks compute mathematical functions, filters, delays, control laws, etc.

Bookmark

`Verification - SUITE`

A marker that allows restoring a simulation cycle previously bookmarked (in a step-by-step simulation only).

Bool

`Design - SUITE/DISPLAY/RAPIDPROTO`

See Boolean.

Boolean

`Design - SUITE/DISPLAY`

A predefined data type in SCADE Suite /a variable type in SCADE Display and Rapid Prototyper (notation: `bool`).

Branch Coverage

`Testing - TEST`

This criteria measures coverage by adding a measurement point on the single exit point of every branch. A branch is defined for each SGFX layer instance, for each OGFX component instance, for each object, and for groups

of consecutive objects having the same priority. Covering the exit point ensures that the entry point of the branch is also covered. Applicable to model coverage with SCADE Display.

Breakpoint

`Design - SUITE`

An intentional stopping or pausing place in the SCADE Suite model, enabled on various control and data elements for debugging purposes. Breakpoints (used with stops conditions) help reaching specific situations during a simulation.

C

Calibration

`Design - SUITE`

The process of fine- tuning a software application on the target or a virtual calibration tool by changing constant parameters that influence the behavior of the regulation algorithms. It is a methodology mainly used in the automotive software development where it has been standardized by the ASAM standard. SCADE Suite supports code generation compliant to the ASAM MCD2 standard suitable for calibration tools.

Call graph

`Design - SUITE`

A view that allows identifying the dependencies between operators.

Caller

`Design - SUITE`

Operator that instantiates another operator.

CAN

`Design - ARCHITECT`

The acronym of Controller Area Network. A broadcast-type bus using a specific message frame format for receiving and transmitting the data.

Causality

`Verification - SUITE`

Causality loops are cyclic dependencies of flow calculation, or a mix of State/ Transition execution and flow calculation.

CDS

`Design - ARINC661`

The acronym for Cockpit Display System.

Certification

`Certification - SUITE/DISPLAY`

From SCADE Suite perspective, two aspects of certification are interesting:

1. **Certification of Software Development or Verification Tools**: SCADE Suite KCG has been certified by TÜV Süd according to the safety standards IEC 61508 (generic) and EN 50128 (Rail) and has been qualified for each SIL. This means that the user can use SCADE Suite and the automatic code generator for safety-related software products development. The user benefits from Certification Credits as listed in the report to the certificate. For Aerospace applications, see Qualification.
2. **Certification of Software products by the user**: The user may face the requirement to deliver a certified software product or a software component that is part of a certified system. His software development process and software product will therefore be subject to a certification process according to the relevant standard (see DO-178C, EN 50128, IEC 61508). Using a software development tool that is qualified for the required standard and safety level, and that comes with Certification credits significantly reduces time and cost to certification.

Certification credits

`Certification - SUITE/DISPLAY`

Acceptance by the certification authority that a process, product or demonstration satisfies a certification requirement. If the credit has been achieved through usage of a tool, it may replace conventional means of achieving the certification requirement.

Certification Kit

`Certification - SUITE/DISPLAY`

A certification kit contains all evidence that is required to achieve qualification of a software tool according to a particular safety standard. It contains certificates, certification reports, and all reports and plans as required.

Certified code generation

`Certification/Generation - SUITE`

SCADE Suite KCG Code Generator has been certified as a product for SIL up to level 3 under IEC 61508 and all safety integrity levels (SILs) for under EN 50128. SCADE Suite KCG generates certified code from the SCADE Suite model-based development system.

Character

`Design - DISPLAY/RAPIDPROTO`

The smallest component of written language that has semantic value. Character refers to the abstract idea, rather than a specific shape (see also Glyph).

Clipping

`Design - DISPLAY/RAPIDPROTO`

A feature to hide the graphical objects or part of graphical objects that are outside an area defined by the user. There are two types of clipping: inside (the outer part of the defined area is hidden) and outside (the inner part of the area is hidden).

Clocked flow

`Design - SUITE`

A flow defined/activated at selected instants. The clock defines the instants when a flow is available.

Co-generation

`Generation - SUITE/DISPLAY`

Automatic generation of integration code between SCADE Suite generated code and SCADE Display generated code, according to the mapping information at model level.

Cockpit Display System

`Design - ARINC661`

The software system responsible for managing the display and input devices.

Compiler Verification Kit (CVK)

`Verification - SUITE`

A tool that enables users to check that the compiler/linker introduces no discrepancy between the Source and the object code.

Condition

`Design - SUITE`

A Boolean expression.

Conditional block

`Design - SUITE`

A control structure to represent branching of control flow. Used in SCADE Suite to control the activation of computations depending on a Boolean condition.

Conditional container

`Design - DISPLAY/RAPIDPROTO`

A symbology object designed to display only one of its children at a given time. It allows displaying an object out of several.

Cone of influence

`Generation - SUITE`

A tree of operators such that the children of an operator are the operators called by this operator. Starting from the root operator, the cone of influence gives all the operators actually used by the model.

Configuration

`Generation - SUITE/DISPLAY/RAPIDPROTO/ARINC661`

Groups a set of parameters associated with a project.

Configuration

`Design - ARCHITECT`

An extension of the SCADE Architect's domain-agnostic metamodel. A configuration defines a new domain specific language (*e.g.*, AADL, AUTOSAR, etc.).

Configuration Management (CM) Gateway

`Management - SUITE/DISPLAY`

A product-specific gateway that allows linking any SCADE Suite model or any SCADE Display specification to the user configuration management system.

Constant expression

`Design - SUITE`

Any expression that can be statically evaluated as a constant. Also called Static expression.

Consumer

`Design - SUITE`

A variable that directly consumes another variable within the same operator (*e.g.*, a variable whose values depend on the selected one). See also Deep consumers.

Context expression

`Design - SUITE`

In a verification project, the design of all property operators must be represented in their context. Context expression consists of creating an observer operator by connecting the system design operator and the property operators.

Continuous control

`Design - SUITE`

Regular periodic computation such as: sampling sensors at regular time intervals, performing signal processing computations on their values, computing control laws and outputting the results. Data is continuously subject to the same transformation.

Continuous simulation

`Verification - SUITE`

In continuous mode, simulation cycles are executed successively. No scenario is required and input values can be entered on the fly.

Contract

`Verification - SUITE`

A specification of expectations before applying a function (or a node) and, supposing they are verified, of some properties on the results of this application. Within the synchronous data flow model of Scade, this contract can be presented as a pair of two observers:

- one corresponding to expectations: the assume that observes the inputs and the past of the outputs.
- one corresponding to ensured properties: the guarantee that observes the inputs and the outputs.

Control activation

`Verification - SUITE`

Predefined integration criterion applying to clocked constructs and measuring whether every state/transition, clocked block, or activate branch is active.

Control formalism

`Design - SUITE`

Control statements (such as SCADE State Machines (SSM), If blocks and When blocks) enabling imperative programming.

Control point

`Design - DISPLAY/RAPIDPROTO`

A point located at each end of a line segment (or on the radius of a circle) and that can be moved to modify the geometric characteristics of an object (its size or its position in the layer).

Co-reporting

`Management - SUITE/DISPLAY`

Automatic co-generation of documentation from a SCADE Suite model and a SCADE Display specification.

Corner case

`Verification - SUITE`

A situation that occurs only outside of the normal functional behavior of the model and usually revealing bugs.

Co-simulation

`Verification - SUITE/DISPLAY`

Model level co-simulation between SCADE Suite and SCADE Display, or between SCADE Suite and an external application (Simulink®, LabVIEW™, FMU-compliant tools, or Ansys Twin Builder). This feature enables to launch at the same time the SCADE Suite Simulation and the SCADE Display Animator or the external application simulator/debugger. Co-Simulation in SCADE Suite can be customized for other applications.

Coverage analysis

`Verification - TEST`

The process of determining the degree to which a proposed software verification process activity satisfies its objectives.

Coverage criterion

`Verification - TEST`

Selectable model coverage metric to use for covering SCADE Suite or SCADE Display models. Available metrics for SCADE Display coverage measures are: Influence, Observable Decision Coverage (ODC), Observable Modified Condition/Decision Coverage (OMC/DC). Available metrics for SCADE Display coverage measures are: Branch Coverage (BC), Decision Coverage (DC), Modified Condition/Decision Coverage (MC/DC).

Coverage observer

`Verification - TEST`

A coverage observer is an operator created for implementing additional coverage points on one or several observed operators without impacting the model. A Tagset utility library containing the operators that introduce the desired coverage points is available to implement observers.

Coverage points

`Verification - TEST`

It is possible to augment structural coverage points used to measure coverage in SCADE Suite models with additional points defined in observer libraries.

Coverage resolution

`Verification - TEST`

Coverage resolution relies upon the justification of uncovered features that reveal implementation and/or verification issues.

CursorPosRequest

`Design - DISPLAY/RAPIDPROTO`

A primitive that allows forcing, under a given condition, the position of a pointing device to the position defined by the primitive.

CVK

`Verification - SUITE`

The acronym of Compiler Verification Kit.

Cycle

`Design - SUITE/DISPLAY`

The logical time used by SCADE Suite when simulating generated C code. It corresponds to one execution of the simulation executable built with SCADE Suite. At each simulation cycle, outputs are computed based on model inputs.

D

Data Structuring

`Design - SUITE`

Declaration of data types, structures, constants and sensors that are defined in the SCADE Suite model.

Data typing

`Design - SUITE`

The Scade language is strongly typed, in the sense that each data flow has a type (Boolean, integer, real, arrays, etc.), and that type consistency in SCADE Suite models is verified by the SCADE Suite tools.

DC

`Verification - TEST`

Acronym for Decision Coverage.

Deactivated code

`Generation - SUITE`

Executable object code (or data) that by design is either (a) not intended to be executed (code) or used (data) (*e.g.*, a part of a previously developed software component), or (b) is only executed (code) or used (data) in certain configurations of the target computer environment (e.g., code that is enabled by a hardware pin selection or software programmed options).

Dead code

`Generation - SUITE`

Executable object code (or data) that, as a result of a design error cannot be executed (code) or used (data) in an operational configuration of the target computer environment and is not traceable to a system or software requirement.

Debugging

`Verification - SUITE`

The process of finding and removing the causes of failures in a SCADE Suite model.

Decision

`Design - SUITE`

A Boolean expression composed of conditions and zero or more Boolean operators. A decision without a Boolean operator is a condition. If a condition appears more than once in a decision, each occurrence is a distinct condition.

Decision Coverage criterion

`Verification - TEST`

This criterion measures coverage by adding a measurement point on each decision to check it has taken the “true” and “false” values. Intermediary between Branch Coverage and MC/DC criteria. Applicable to model coverage with SCADE Display.

Declaration

`Design - SUITE`

When designing your model in SCADE Suite, you first have to create the building blocks of your design. This is the declaration phase in your model design. See also Design instance.

Deep consumer

`Design - SUITE`

A variable that consumes another variable within the same operator or outside (*e.g.*, across operator calls).

Deep producer

`Design - SUITE`

A variable that produces another variable within the same operator or outside (*e.g.*, across operator calls).

Definition File

`Design - ARINC661`

A data format for the description of the user interface elements.

Definition parameter

`Design - ARINC661`

Parameters of the A661 widgets. Each parameter is unique in the layer and editable (except ParentIdent).

Definition file (DF)

`Design - ARINC661`

A file associated with one UA. The DF contains the UA Layer Description (UALD) to be displayed on the CDS.

Dependencies

`Design - SUITE`

See Call graph.

Descender

`Design - DISPLAY/RAPIDPROTO`

The portion of some lowercase letters, such as g and y, that extends or descends below the baseline.

Design capture

`Design - SUITE`

Capture of architecture and algorithm design developed, for example, in SysML/UML or Simulink® environments, to be translated into SCADE Suite models.

Design instance

`Design - SUITE`

Graphical instantiation of design elements in Network Views that represent calls or references to design elements.

Design Verifier (DV)

`Verification - SUITE`

A formal proof engine within the SCADE Suite tool that uses formal methods to prove the properties of a Scade design. It enables to prove that a design is correct with respect to expressed properties and allows finding bugs very early in the design cycle.

Determinism

`Certification - SUITE`

The same sequence of inputs always produces the same sequence of outputs.

Deterministic model

`Certification - SUITE`

A model that produces defined values, in the sense that for given input sequences, the output sequences are completely defined.

DF

`Design - ARINC661`

The acronym for Definition File.

Diagram

`Design - SUITE`

See Textual diagram or Graphical diagram.

Discrete control

`Design - SUITE`

Changing behavior according to external events originating either from discrete sensors and user inputs or from internal program events, for example, value threshold detection. Discrete control is used when the behavior varies qualitatively as a response to events. This is characteristic of modal human-machine interface, alarm handling, complex functioning mode handling, or communication protocols.

Display Unit

`Design - ARINC661`

The physical device that displays the graphic (a screen).

DO-178C

`Certification - SUITE/DISPLAY`

DO-178C/ED-12C was first published in 2011 by RTCA (Requirements and Technical Concepts for Aviation) and EUROCAE (a non-profit organization addressing aeronautic technical problems). It was written by a group of experts from aircraft and aircraft equipment manufacturing companies and from certification authorities. It provides guidelines for the production of software for airborne systems and equipment. The objective of the guidelines is to ensure that software performs its intended function with a level of confidence in safety that complies with airworthiness requirements. These guidelines specify:

- Objectives for software life-cycle processes
- Description of activities and design considerations for achieving those objectives
- Description of the evidence indicating that the objectives have been satisfied

Double simulation

`Verification - SUITE`

Capability to run a simulation session on two similar models at the same time.

DU

`Design - ARINC661`

The acronym of Display Unit.

DV

`Verification - SUITE`

The acronym of Design Verifier.

E

EMF

`Design - DISPLAY`

Eclipse Modeling Framework. See also SCADE Display EMF API.

EN 50128

`Certification - SUITE`

This European Standard applies to all software used in development and implementation of railway control and protection systems. It specifies procedures and technical requirements for the development of programmable electronic systems for use in railway control and protection applications. It is aimed at use in any area where there are safety implications.

Encapsulation

`Design - SUITE`

A structuring mechanism that allows grouping several design elements into a new operator, state, or state machine.

Enumeration

`Design - SUITE`

A predefined data type in SCADE Suite consisting of a set of named values called elements.

Event

`Design - ARINC661`

Notification sent by the CDS to a UA to indicate a crew member interaction has occurred on a widget owned by that UA.

Environment

`Design - DISPLAY`

A SCADE Display environment is defined by the used graphical library (*e.g.*, OpenGL), the subset of graphic objects (primitives, masks, advanced objects, etc.), the subset of layer types available and the list of preferences and options available in the GUI.

Equation Set

`Design - SUITE`

A construct assembling design elements together as groups of equations to enable global commenting, annotation, or tracing whole equation sets to requirements. In diagrams, equation sets can be visually identified by applying graphical styles.

Error

`Verification - SUITE/DISPLAY`

Discrepancy between a computed, observed or measured value or condition and the true, specified or theoretically correct value or condition.

F

Failure

`Verification - SUITE/DISPLAY`

Termination of the capability of a functional unit to perform a required function.

Fault

`Verification - SUITE/DISPLAY`

Abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function.

Fault tolerance

`Verification - SUITE/DISPLAY`

The ability of a functional unit to continue to perform a required function in the presence of faults or errors.

Final state

`Design - SUITE`

Used to mark a state for synchronization.

Flat shading

`Design - DISPLAY`

A feature to display all the points in a polygon with a single color.

FMI

`Design - SUITE/DISPLAY/RAPIDPROTO`

The acronym for Functional Mock-up Interface.

FMU

`Design - SUITE/DISPLAY/RAPIDPROTO`

The acronym for Functional Mock-up Unit.

Fold

`Design - SUITE`

A Scade high-level operator to call multiple instances of a given Scade operator in an iterative way. Inputs of the Fold are the vectors of N inputs to distribute over the different instances plus a special input called the accumulator. The iterated operator must be design such that it dedicates its first input for the accumulator from the previous instance, and its first output for the next instance. The Fold operator ensures passing automatically the accumulator from an instance to the other. The output of the Fold is the accumulated results provided by the last instance call.

Font

`Design - DISPLAY/RAPIDPROTO`

A collection of glyphs, generally with at least one glyph associated with each character in the font's character set.

Forked transition

`Design - SUITE`

Allows representing decisions branches based on conditions and priorities in a SCADE State Machine.

Formal methods

`Design - SUITE`

Descriptive notations and analytical methods used to construct, develop, and reason about mathematical models of system behavior.

Formal verification

`Verification - SUITE`

Support of exhaustive verification of critical properties, using Design Verifier.

Full screen

`Verification - DISPLAY/RAPIDPROTO`

A function used to display the specification on all the screen. All window decorations are then hidden.

Function

`Design - SUITE`

A stateless operator that contains no internal memory and defined in graphical or textual diagrams.

Functional Mock-up Interface

`Design - SUITE/DISPLAY/RAPIDPROTO`

A tool independent standard for the exchange of dynamic models and for Co-Simulation. SCADE products support FMI export for Model Exchange. For detailed information, visit Modelisar site. See also Functional Mock-up Unit.

Functional Mock-up Unit

`Design - SUITE/DISPLAY/RAPIDPROTO`

A component which implements the FMI standard. The FMU instance is driven by the FMI-compliant modeling and simulation environment. See also Functional Mock-up Interface.

G

Gateway

`Management - SUITE/DISPLAY`

A module that allows connecting SCADE Suite or SCADE Display environments to external tools.

Generic type

`Design - SUITE`

Generic types are unspecified types that can be substituted for any type at instantiation of operator calls. They are expressed locally on any input, output, or local variable of an operator. See Polymorphic operator.

Graphical diagram

`Design - SUITE`

The graphical representation of an operator design displayed within Network views.

Graphical panel

`Verification - SUITE`

A graphical panel constructed from a library of widgets for running interactive simulation sessions of your SCADE Suite model.

Glyph

`Design - DISPLAY/RAPIDPROTO`

The actual shape (bit pattern, outline) of a character image. For example, an italic 'a' and a roman 'a' are two different glyphs representing the same underlying character. In this strict sense, any two images which differ in shape constitute different glyphs.

Graphical formalism

`Design - SUITE`

SCADE Suite uses high-level graphical notations for modeling: data flow blocks and state machines. The graphical formalism of state machines allows to model the behavior of reactive systems with their control logic environment.

Grid

`Design - DISPLAY/RAPIDPROTO`

A feature that allows positioning primitives precisely through mouse positioning thanks to the grid attraction.

Group

`Design - DISPLAY/RAPIDPROTO`

A set of objects defined as children in a basic container.

Guarantee

`Design - SUITE`

One of both observers that characterize a contract. The guarantee expresses the contract's post-conditions by observing the inputs and the outputs. Under the assumption that the assume conditions are true, the operator must ensure that the guarantee conditions are true.

Guides

`Design - DISPLAY/RAPIDPROTO`

Graphical vertical and horizontal line marks positioned by user on the Visualization area. This feature allows positioning and aligning primitives precisely by the help of the guides.

H

Haloing

`Design - DISPLAY/RAPIDPROTO`

A feature to create black outlines for lines.

Hardware/software integration

`System - SUITE/DISPLAY`

The process of combining the software into the target computer.

Hidden input

`Design - SUITE`

An operator graphical property used to improve model readability by reducing the number of links. It allows entering directly the input value.

Hierarchy

Design - **SUITE**

Hierarchical decomposition of block diagrams and state machines for designing the system architecture.

High-level requirements (HLR)

Certification - **SUITE**/**DISPLAY**

Software requirements developed from analysis of system requirements, safety-related requirements, and system architecture.

Higher-order operator

Design - **SUITE**

A user-defined or predefined operator on which a pattern can be applied so that it can then be used in its scope as any primitive operator to build expressions (*e.g.*, iterators, Boolean activation, etc.).

History connector

Design - **SUITE**

A transition connector that allows resuming the execution of a state in a SCADE State Machine as it was when last exited and executed.

Host computer

SUITE/**DISPLAY**

The computer on which the software is developed.

HTML

SUITE/**DISPLAY**

SCADE Suite is able to generate project reports to document a design at anytime in HTML (or RTF format). Hyperlinks allow navigating easily in HTML reports.

I

IEC 61508

Certification - **SUITE**

IEC is the International Electrotechnical Commission. IEC 61508 is a standard that contains requirements to minimize failures in electrical/electronic/ programmable electronic safety-related systems.

Imported constant

Design - **SUITE**

Externally defined user constants can be introduced in SCADE Suite models as imported constants. Constants can be typed with SCADE Suite or imported types.

Imported operators

Design - **SUITE**

Operators (nodes /functions) defined with external C code to extend Scade language expressiveness.

Imported type

Design - **SUITE**

Externally defined user types can be introduced in SCADE Suite models as imported types. Only imported operators can process the actual data of imported typed flows.

Independence

Certification - **SUITE**/**DISPLAY**

Separation of responsibilities that ensures the accomplishment of objective evaluation. (1) For software verification process activities, independence is achieved when the verification activity is performed by a person(s) other

than the developer of the item being verified, and a tool(s) may be used to achieve an equivalence to the human verification activity. (2) For the software quality assurance process, independence also includes the authority to ensure corrective action.

Influence criterion

`Verification - TEST`

This criterion measures coverage based on tags attached to data flows of the model and on tags related to the activation of scopes introduced by control structures (state machines and conditional activation operators). With this criterion, Boolean primitives behave as any combinatorial primitive by always propagating the tags present on the inputs to the outputs regardless of the actual Boolean value of the streams. Applicable to model coverage with SCADE Suite.

Initial state

`Design - SUITE`

The active state whenever a state machine is started.

Input

`Design - SUITE`

An operator interface. An input is an entrance inserted into an operator to activate/modify a process.

Instantiation

`Design - SUITE`

An object is instantiated when it is used within an equation or a diagram. Instantiation is opposed to declaration.

Instrumentation

`Design - TEST`

Production of all needed data from SCADE Suite or SCADE Display models before launching coverage acquisition.

Integer

`Design - SUITE/DISPLAY/RAPIDPROTO`

A predefined data type in SCADE Suite /a variable type in SCADE Display and Rapid Prototyper (notation: `int`).

Integral process

`Certification - SUITE/DISPLAY`

A process that assists the software development, processes, and other integral processes and, therefore, remains active throughout the software life cycle. The integral processes are the software verification process, the software quality assurance process, the software configuration management process, and the certification liaison process.

Iterator

`Design - SUITE`

Point-wise application of operators enabling iterative processing of arrays. The Scade language provides predefined ways to apply an operator to arguments belonging to an array data type. These operators successively apply this operator on every element of the array and produce either a new array (the map family) or an element of the basic type of the array (the fold family).

J

JPEG

`Design - SUITE/DISPLAY/RAPIDPROTO`

A standard of compression for photographic images. Such image files can be used in SCADE Suite to edit operator symbols and in SCADE Display or Rapid Prototyper to be set as background.

Justification

`Verification - TEST`

Whenever coverage cases are not sufficient to cover software implementation, it is possible to force coverage by textual justification and save corresponding justification records with coverage results.

K

KCG Code Generators

`Generation - SUITE/DISPLAY`

The qualifiable SCADE Suite KCG code generator and SCADE Display KCG code generator produce simple C code that has all the properties required for safety-critical embedded software and display applications.

Kiviat diagram

`Management - SUITE`

A graphical depiction of KCG metrics allowing to quickly identify the metrics to focus on for a particular project. Each metric is represented as a spoke of a wheel. The fractional value of each metric at a certain time, *e.g.*, 75/100, determines a point on each spoke. Those points then determine the vertices of a polygon whose size and shape indicate the relative status of the project.

L

Layer

`Design - DISPLAY/RAPIDPROTO/ARINC661`

A layer is composed of a set of objects and has a transparent background. Superimposing layers allows to display complex images.

In the context of ARINC 661, a layer is a collection of widgets controlled by the same UA, described in a DF.

Legacy code

`Generation - SUITE`

Source code already existing and implemented in another programming language that must be used in SCADE Suite models (*e.g.,* as library project). SCADE Suite allows implementing imported code within SCADE Suite models. Once imported code is correctly implemented and its source files properly declared, the corresponding imported entities can be used in simulation of SCADE Suite models.

Library

`Design - SUITE`

SCADE Suite provides a number of example libraries at project creation. These libraries have the following properties:

- Contain complex user-defined operators
- Fully tested, using well-known symbols
- Fully documented
- Generic when appropriate

Libraries may be created to structure user definitions in SCADE Suite projects and facilitate reuse and sharing. Libraries can contain only user-defined operators.

Library operator

`Design - SUITE`

An operator defined within a SCADE Suite library and that can be shared between several SCADE Suite models.

Line loop

`Generation - DISPLAY`

A line drawing mode used to draw lines from vertices: each vertex is linked to the next vertex by a line and the last vertex is linked to the first one.

Line segment

`Design - DISPLAY/RAPIDPROTO`

A line connecting two points.

Line stipple

`Design - DISPLAY/RAPIDPROTO`

A one-dimensional pattern defining the graphical aspect of a line: this pattern is repeated along the line and is described by drawn and non drawn segments. The segments have fixed length in pixels.

Line strip

`Generation - DISPLAY`

A line drawing mode used to draw lines from vertices: each vertex is linked to the next vertex by a line but unlike the line loop mode, the last vertex is not linked to the first one.

Local variable

`Design - SUITE`

Variable seen only by the operator in which it is declared. Can be used as many times as necessary.

Look and Feel

`Design - ARINC661`

The graphical characteristics of a widget, which are not managed by a UA but by the CDS to insure a homogeneous HMI. This terminology also applies to widget behavior internal to the CDS, leading to a state diagram internal to the CDS that manages transition between visual representations.

LookUp-Table (LUT)

`Design - SUITE`

A data structure, usually an array or associative array, often used to replace a runtime computation with a simpler array indexing operation.

Low-level requirements (LLR)

`Management - SUITE/DISPLAY`

Software requirements derived from high-level requirements, derived requirements, and design constraints from which source code can be directly implemented without further information.

M

Macrostate

`Design - SUITE`

State in which you can nest SCADE State Machines gathering multiple other states, as well as data flows.

Magnetism

`Design - DISPLAY/RAPIDPROTO`

Determines how close an object or a control points must be to a guide, to another object or to the grid in order to be snapped onto it.

Map

`Design - SUITE`

Scade high-level operator to call multiple instances of a given Scade operator in an iterative way. Inputs of the Map are the vectors of N inputs to distribute over the different instances. The output of the Map is the vector of outputs provided by each instance.

Mapi

`Design - SUITE`

A map iterator with access to the index.

Mapping groups

`Verification - SUITE`

A method provided by Design Verifier to handle complex properties and redefine part of a model design.

Matrix

`Management - SUITE`

A matrix is encoded by an array of arrays.

MC/DC

`Verification - TEST`

Acronym for Modified Condition/Decision Coverage.

Metamodel

`Design - SUITE`

SCADE Suite includes metamodels (Project, Editor, Model Coverage, Annotation, Scade Language, Scade Graphics) that can be used with the Script Wizard to extract data from SCADE Suite projects.

MicroC

`Generation - SUITE`

SCADE Suite KCG Code Generator can generate code for use with µC/OS-II Real Time Operating System (RTOS) from Micriμm Technologies Corporation.

Model-Based Design (MBD)

`Design - SUITE`

SCADE Suite is a model-based design tool. In SCADE Suite, the model is the reference, not an approximation of reality. The adoption of Model-Based Design enables to:

- address the complexity inherent in control system designs
- start software design before physical systems are available
- verify the system prior to implementation, thus to facilitate the detection and elimination of errors in requirements specification significantly earlier in the development cycle

- create a structure for software reuse that permits a reliable and cost effective upgrade path for established designs

Model Coverage for SCADE Display

`Verification - TEST`

A SCADE Test module that measures the model and code coverage of SCADE Display models reached by a high-level requirements-based test suite.

Model Coverage for SCADE Suite

`Verification - TEST`

A SCADE Test tool that measures the model coverage of SCADE Suite models (with code coverage implication for SCADE Suite KCG-generated code) reached by a high-level requirements-based test suite.

Model Coverage analysis

`Verification - TEST`

The process of determining the degree to which a proposed software verification process activity satisfies its objective and fully covers the model.

Model Coverage batch mode

`Verification - TEST`

The batch mode is a certified chain dedicated to compute the final coverage and produce coverage analysis report for certification activities. SCADE Test Model Coverage for SCADE Suite is qualified as a verification tool and only this batch mode can be used for certification purposes.

Model Coverage interactive mode

`Verification - TEST`

The interactive mode facilitates the understanding of coverage cases, the visualization of coverage results, and the justification of corner cases at model level.

Model Diff

`Verification - SUITE`

A SCADE Suite tool that enables the user to analyze and display the semantic differences between two SCADE Suite models. Model Diff can be used to compare two different models or two versions of the same models, for instance, to highlight the modifications made to a model.

Model inconsistency

`Verification - SUITE`

See SCADE Suite Checker.

Model-Level coverage measurement

`Verification - TEST`

Model-level coverage verifies that each element in the SCADE Suite or SCADE Display model is dynamically activated during simulation with respect to selected coverage criteria. SCADE Test Model Coverage batch tool computes measurements of coverage at model level.

Modelview matrix

`Generation - DISPLAY`

A 4x4 matrix that transforms the coordinates of the graphic objects expressed in the user coordinates system, into the pixel coordinates system adapted to the viewport. See also Orthographic Matrix.

Modified Condition/Decision Coverage criterion

`Verification - TEST`

This criterion measures coverage by adding measurements on each vector of conditions composing a decision to check that two vectors independently influence the decision value. Applicable to model coverage with SCADE Display.

Monomorphic operator

`Generation - SUITE`

Any operator which is not parameterized by size or which has a specific type. See also Generic type and Polymorphic operator.

N

NPlicator Container

`Design - DISPLAY/RAPIDPROTO`

A container that allows manipulating arrays of objects, with the capability to manage in run-time the associated plugs individually. Plugs manipulated within NPlicator containers are of array type.

Nested state machine

`Design - SUITE`

A state machine instantiated within another state machine. This is an easy way to model control structures and states hierarchy.

Node

`Design - SUITE`

Operator with memory and defined in graphical or textual diagrams.

O

Object code

`Generation - SUITE`

The code produced by a compiler.

OBS file

`Verification - SUITE`

File that provides the list of variables to be observed in addition to the inputs/outputs of the root node and any operator instance it contains.

Observer operator

`Verification - SUITE`

An operator that contains both a system design and the property operator correctly connected to the inputs/outputs of the system design. In a design verification project, the observer operator is required along with the property operator to prove any property with Design Verifier.

ODC

`Verification - TEST`

Acronym for Observable Decision Coverage.

Observable Decision Coverage criterion

`Verification - TEST`

This criterion measures coverage based on tags able to distinguish between the influence of True and the influence of False for the monitoring of Boolean flows and on tags introduced to capture the range of indexes used in dynamic projections. With this criterion, the propagation rules for Boolean primitives is the same as for Influence. The semantics of tag propagation of this criterion ignores the MC/DC masking effect of Boolean flows on coverage measurements. Applicable to model coverage with SCADE Suite.

Observable Modified Condition/Decision Coverage criterion

`Verification - TEST`

This criterion measures coverage based on the same tags as ODC and a semantics of tag propagation that takes into account the masking effect over coverage measurements. Applicable to model coverage with SCADE Suite.

OGF/OGFX file

Design - DISPLAY

A file describing a part of a display specification and that can be used as a reference object.

OGLX

Design - DISPLAY

The SCADE Display graphics library supporting OpenGL ES-SC safety-critical graphic features (including vectorial drawings, complex masks, bitmaps and texture-mapping, haloing and anti-aliasing support).

OID

Design - SUITE

Object unique identifier (sometimes used internally to identify objects).

OMC/DC

Verification - TEST

Acronym for Observable Modified Condition/Decision Coverage.

Open System Description

Design - ARCHITECT/SUITE

A format developed in the context of the “Smart, Safe & Secure Platform” (S3P) project (see http://www.s3p-alliance.fr/S3P-en.html).

OpenGL ES-SC

Design - DISPLAY

A profile of the OpenGL standard and supported by the SCADE Display OGLX graphics library. This profile is defined by the Khronos industry consortium to provide a robust set of graphics functionality to embedded, safety critical systems.

OpenType

Design - DISPLAY/RAPIDPROTO

A type of font that extends the TrueType font format and that can contain PostScript data.

Operator

Design - SUITE

The functional elements identified in the model high-level requirements. Operators help designing the architecture of models graphically. SCADE Suite supports the following operators: nodes, functions, imported nodes/ functions, predefined operators, library operators. All types of SCADE Suite operators (except built-in predefined operators) can be defined with the support of graphical and textual diagrams.

An operator is represented as a block diagram and is seen as a black box and is connected through its formal I/O interface.

Operator parameterized by size

Design - SUITE

Any operator whose effective definition depends on one or several “size parameters”.

Origin

Design - DISPLAY/RAPIDPROTO

The point on a baseline used as a reference location for drawing a glyph.

Orthographic Matrix

Generation - DISPLAY

A 4x4 matrix used to initialize the modelview matrix. The modelview matrix can be transformed through transformation functions (translation, rotation, scaling) and restored to its initial value thanks to a function.

OS

`Design - SUITE/DISPLAY`

The acronym of Operating System.

OSD

`Design - ARCHITECT/SUITE`

See "Open System Description".

OSEK/VDX

`Generation - SUITE`

An industry standard for an open-ended architecture on distributed control units in vehicles. It describes three areas: communications, network management and operating system. SCADE Suite KCG Code Generator can generate code for use with OSEK Real Time Operating System (RTOS).

Output

`Design - SUITE/DISPLAY/RAPIDPROTO`

An operator interface. An output is an exit or changes which exits an operator and which activate/modify a process.

Overflow

`Verification - SUITE`

A situation where the maximum representation capability of a data storage is lower than a given value it is supposed to store. For example, adding one to 255 on a 8 bits unsigned integer storage induces an overflow.

P

Packages

`Design - SUITE`

Are the SCADE objects that allow you to define namespaces where you can group model elements together. These elements can be shared and reused through out other SCADE models and projects.

Panel container

`Design - DISPLAY/RAPIDPROTO`

A rectangular container with size (SizeX, SizeY). All children outside the container boundaries are clipped.

Parallelism

`Design - SUITE`

Parallel composition of several SCADE State Machines or block diagrams for modeling concurrent behaviors of the system.

Parameter

`Design - ARINC661`

A named value attached to a widget. For each widget type the standard specifies a collection of parameters with their name and types.

Partial iterators

`Design - SUITE`

Iterators of which application is restricted to the first items of their arguments until a certain condition is met. Partial map notation is `mapw` and partial fold notation is `foldw`.

PGF file

`Management - DISPLAY`

Any node of the structure tree that is put under configuration management is stored in a PGF File on which the parent file is pointing.

Picture table

`Design - ARINC661`

A table which defines a list of pictures, to be generated in a specific format.

Pin

`Design - SUITE`

Pins enable to connect the various data flow design elements present in the active Network View. It is possible to show/hide the rank of operator call pins and the name of inputs and outputs beside each pin of operator calls.

Pitch

`Design - DISPLAY/RAPIDPROTO`

The size (width/height ratio) in user unit/pixels of a layer/specification.

Pixel coordinates

`Generation - DISPLAY`

Coordinates expressed directly in pixels.

Plug

`Design - DISPLAY/RAPIDPROTO`

A variable connected to a symbology object property. This type of connection allows modifying the object's properties (position, visibility, color, etc.) from a dynamic point of view.

PNG

`Design - SUITE/DISPLAY/RAPIDPROTO`

A bitmapped image format that employs lossless data compression.

Polygon

`Generation - DISPLAY`

A shape drawing mode used to draw a shape from vertices: the border of the shape is defined by the vertices sequence.

Polygon smooth

`Design - DISPLAY/RAPIDPROTO`

See Anti-aliasing.

Polyline

`Design - SUITE`

A line style of transitions in state machines.

Polymorphic operator

`Design - SUITE`

Any operator in which one or several data types are left unspecified and determined by the calling context. Such data types are defined as generic types (' T notation) on any variable (input, output, local variable, probe) of the operator. See also Generic type and Monomorphic operator.

Polymorphism

`Design - SUITE`

A principle of typed programming languages allowing the definition of functional operators interface with generic types. For example, the Scade native operator + is polymorphic as it can be applied on integer or real numbers.

Portable code

`Generation - SUITE/DISPLAY`

Independent from the target and the system scheduler.

Pragmas

`Verification - SUITE`

Free text information attached to Scade modeling objects and passed to the C code as comments.

Predefined operator

`Design - SUITE`

Built-in operator that provides access to basic primitives. See User-defined operator.

Preemption

`Design - SUITE`

The ability to interrupt or allow state transitions for handling exceptional situations in the system.

Primitive

`Design - DISPLAY/RAPIDPROTO`

A basic figure component (a point, a line segment, a polygon, an arc, etc.) defined by vertices.

Primitive definition mode

`Generation - DISPLAY`

A sequence of commands executed between sglBegin and sglEnd commands.

Producer

`Design - SUITE`

A variable that directly produces another variable within the same operator (*e.g.*, all variable whose values produce the selected one).

Primitive

`Design - DISPLAY/RAPIDPROTO/ARINC661`

A basic figure component (a point, a line segment, a polygon, an arc, etc.) defined by vertices.

Proof

`Verification - SUITE`

See Proving.

Property operator

`Verification - SUITE`

An operator with inputs representing the data needed to express the property and with a single Boolean output combining these inputs. In a verification project, the property operator is required along with the observer operator to prove an operator property.

Proving

`Verification - SUITE`

In a verification project, Design Verifier can be used to prove the validity of formal properties that you specify.

Q

Qualification

`Certification - SUITE/DISPLAY`

A software development or verification tool needs to be qualified in order to be integrated into the development process for a certified software product. Qualification is the process of achieving this objective. Depending on the relevant standard, it may be based on tool certification and/ or a certification kit.

Quality Assurance (QA) process

`Certification - SUITE/DISPLAY`

Provide confidence that the software life cycle processes produce software that conforms to its requirements by assuring that the processes used to develop the software are compliant with the applicable quality management system requirements and certification standards (DO-178C, EN 50128, IEC 61508, ISO 26262) and are effectively implemented, that deficiencies are detected, recorded and resolved. This is done through reviews and audits.

R

Rapid Prototyper panel

`Design - SUITE`

See Graphical panel.

Rapid prototyping

`Design - SUITE`

Rapid Prototyping enables to confront a SCADE Suite model with a real physical system. See SCADE Suite Gateway for LabVIEW.

Ratio user unit/pixels

`Design - DISPLAY/RAPIDPROTO`

Relation between the size of a layer in user unit and its size in pixels.

Real type

`Design - SUITE/DISPLAY/RAPIDPROTO`

A Scade predefined type.

Reference object

`Design - DISPLAY/RAPIDPROTO`

A part of a model saved as an object in an OGFX file (SCADE Display) or WGFX file (Rapid Prototyper). Once imported into a model through a reference container, all the plugged variables are seen as object properties and pluggable properties.

Requirements traceability

`Management - SUITE/DISPLAY`

The objective of requirements traceability is to ensure that all requirements can be shown as properly met.

Review

`Certification - SUITE/DISPLAY`

The act of inspecting or examining software life cycle data, software project progress and records, and other evidence to assess compliance with RTCA/ DO-178C objectives. Review is an encompassing term and may consist of a combination of reading documents, interviewing project personnel,

witnessing activities, sampling data, and participating in briefings. A review may be conducted at your own desk, at an applicant's facility, or at an applicant's supplier's facility.

RGB 565

`Design - DISPLAY/RAPIDPROTO`

A color format. 16 bits are used to specify the color value: 5 bits to specify the red component (value between 0 and 31), 6 bits to specify the green component (value between 0 and 63), and 5 bits to specify the blue component (value between 0 and 31).

RGB 888

`Design - DISPLAY/RAPIDPROTO`

A color format. 3x8 bits are used to specify the color: each color component, red, green and blue, is coded by 8 bits (value between 0 and 255).

RGFX file

`Design - RAPIDPROTO`

A file that corresponds to a graphical panel.

Risk

`Verification - SUITE`

Combination of the frequency, or probability, and the consequence of a specified hazardous event.

Robustness

`Verification - SUITE/DISPLAY`

The extent to which software can continue to operate correctly despite invalid inputs.

RTOS

`Certification - SUITE`

The acronym of Real-Time Operating System.

RTOS Adaptors

`Generation - SUITE`

A code adaptor that extends the code generation capabilities of the product family to other targets and platforms.

S

SCADE State Machine (SSM)

`Design - SUITE`

A Scade hierarchical state machine commonly used for modeling intuitively the behavior of an application based on control flow and decision logic. A SCADE State Machine, in a given state, reacts to various signals through transitions. These transitions are triggered depending on some conditions and activate other states. Priorities on transitions enforce model determinism.

Safety Critical

`Certification - SUITE/DISPLAY`

Freedom from unacceptable levels of risk.

Safety integrity levels (SIL)

`Certification - SUITE/DISPLAY`

Classification number which determines the techniques and measures that have to be applied in order to reduce residual software faults to an appropriate level.

Safety-related software

`Certification - SUITE/DISPLAY`

Software which carries responsibility for safety.

SCADE Display

`Design - DISPLAY`

The Safety Critical Application Development Environment dedicated to embedded display systems.

SCADE Display Animator

`Verification - DISPLAY`

A module that allows to animate a GUI specification using simple predefined scenarios without writing any line of code. It enables different modes of full screen animation.

SCADE Display EMF API

`Design - DISPLAY`

A stand-alone read/write Java API providing an interface for reading and writing SCADE Display models through an EMF API.

SCADE Display Design Checker

`Verification - DISPLAY`

A module that allows checking a display design efficiency. It optimizes the performances of the display specification and its generated code by checking design and the design rules right at the specification level.

SCADE Display Editor

`Design - DISPLAY`

A module that provides a set of graphic primitives and a state of the art interface to edit them. It enables a detailed specification of the display and supports vector drawings, bitmap textures, full RGB and Alpha-blending.

SCADE Display KCG Code Generator

`Generation - DISPLAY`

A code generator developed in compliance with the objectives of various safety standards (DO-178C, EN 50128, IEC 61508, or ISO 26262) and distributed with certification kits of the corresponding standards. It generates safe and retargetable C code directly from the display specification, thus enabling developers to eliminate low-level testing activities.

SCADE files

`Design - SUITE`

Files that contain design definitions in textual format. SCADE files can be edited directly from SCADE Suite Textual Editor or outside SCADE Suite as any text file.

SCADE LifeCycle Reporter

`Management - ARCHITECT/SUITE/DISPLAY/ARINC661`

A tool integrated to SCADE products and that provides automatic documentation generation. SCADE LifeCycle Reporter generates customizable reports in RTF or HTML format.

SCADE Rapid Prototyper

`Design - RAPIDPROTO`

A product that enables and facilitates the design of interactive graphical panels and the usage of graphical panels for model simulation and cross-team sharing.

SCADE LifeCycle ALM Gateway

`Management - ARCHITECT/SUITE/DISPLAY/ARINC661`

A gateway that provides SCADE products with the ability to connect to ALM tools and create traceability links between any SCADE constructs and requirements to perform traceability analysis in ALM tool environments.

SCADE Suite

`Design - SUITE`

The Safety Critical Application Development Environment dedicated to the production of safety-critical embedded software.

SCADE Suite Checker

`Verification - SUITE`

A tool integrated to SCADE Suite Editor that performs statically semantics checks of a SCADE Suite model and generates an interactive error report.

SCADE Suite Editor

`Design - SUITE`

A module that provides a set of integrated features which provides a clear and comprehensive representation of the project framework based on efficient graphic designing and multi-view management, compliant with the SCADE Suite methodology.

SCADE Suite Gateway for VeriStand™

`Verification - SUITE`

A gateway that enables the co-simulation of SCADE Suite models in the National Instruments VeriStand environment to verify complex algorithms. This gives access to all VeriStand instrumentation capabilities to reproduce the application environment, including physical data acquisition capabilities.

SCADE Suite Simulink® Importer

`Verification - SUITE`

A tool integrated to SCADE Suite Editor that translates into SCADE Suite models the algorithm design from Simulink models, Stateflow charts, and MATLAB variables.

SCADE Suite KCG Code Generator

`Generation - SUITE`

A code generator developed in compliance with the objectives of various safety standards (DO-178C, EN 50128, IEC 61508, or ISO 26262) and distributed with certification kits of the corresponding standards.

SCADE Suite project

`Management - SUITE`

A project is included into a workspace and several projects can be grouped together in the same workspace. SCADE Suite projects are required for developing SCADE Suite models, testing SCADE Suite models, or verifying SCADE Suite models with formal proofs from Design Verifier.

SCADE Suite Simulation

`Verification - SUITE`

A module that enables the simulation and debugging of SCADE models by executing C code which is transparently generated by Code Generator.

SCADE Automotive Package

`Design - ARCHITECT`

A SCADE Suite-bundled module that extends and simplifies system design capabilities for the Automotive market. It includes an AUTOSAR-dedicated SCADE Architect configuration and import/export capabilities compliant with the AUTOSAR standard. SCADE Suite includes an AUTOSAR portable component generator.

SCADE Avionics Package

`Design - ARCHITECT`

A solution that extends and simplifies system design capabilities for the Aerospace and Defense (A&D) market. It includes several SCADE Architect configurations (Avionics, A429, AFDX, CAN, A653, and FACE) and import/export capabilities compliant with the FACE standard. SCADE Suite includes a FACE portable component generator.

SCADE Architect Checker

`Verification - ARCHITECT`

A tool integrated to SCADE Architect Editor, that checks SCADE Architect models for predefined or project-dependent constraints with a set of selectable rules. Detected problems are reported in interactive reports and some can be resolved by applying quick fixes. Custom rules can be created in Java, OCL, or TCL formats.

SCADE Test project

`Testing - TEST/SUITE/DISPLAY`

A project created for describing test procedures and test cases in SCADE Test Environment. A test project is always associated with the SCADE Suite or SCADE Display project to be tested and usually requires a separate test result project.

SCADE UA DF Generator

`Generation - ARINC661`

A module that enables to generate Definition Files (DF) in binary and XML format from any valid SCADE Display ARINC 661 model designed with the UA Page Creator, according to the A661 XML configuration files, and conform to ARINC 661-4 specification.

SCADE UA Page Creator

`Design - ARINC661`

A module that enables designing ARINC 661 UA pages by instantiation and parameterization of ARINC 661 widgets.

SCADE Server Creator

`Design - ARINC661`

A module that enables creating and editing the A661 XML configuration files that define the ARINC 661 Widget Library.

Scenario

`Verification - SUITE/DISPLAY`

Provides a memory snapshot of the simulation/animation session at a given cycle.

SCT file

`Design - DISPLAY/RAPIDPROTO`

A file in which is stored the specification color table.

SDT file

`Management - DISPLAY`

A file in which are stored the Reporter generation settings.

Semantic analysis

`Verification - SUITE`

Analysis according to a set of semantics rules. In Scade, semantic analysis relies on the Scade formal semantics and used to check the correctness of a Scade design with respect to this semantics such as design completeness, type checking, correct initialization of flows etc.

Semantic checks

`Verification - SUITE`

Checks related to the Scade language and performed by SCADE Suite KCG Code Generator (Post-syntax analysis, Namespace analysis, Type analysis, Clock analysis, Causality analysis and Initialization analysis).

SGF/SGFX file

`Design - DISPLAY`

A file that corresponds to a display specification.

Shading

`Design - DISPLAY/RAPIDPROTO`

When a line or a polygon is drawn, color values are specified for corresponding pixels. There are two shading types: flat shading and smooth shading.

Signal

`Design - SUITE`

A flow which is present from time to time with a particular clock.

SIL

`Certification - SUITE/DISPLAY`

The acronym of Safety Integrity Levels.

Simulation

`Verification - SUITE`

SCADE Suite Simulation provides an environment in which to simulate and test the model (which is also the code to embed) in continuous, interactive and batch modes. Simulation is thus the first debugging phase of the model.

Smooth shading

`Design - DISPLAY/RAPIDPROTO`

On a polygon, displays the points in a polygon with smoothly changing colors across the contour of the polygon. Color values are interpolated so that adjacent pixels appear continuous.

Software

`Design - SUITE/DISPLAY`

Intellectual creation comprising the programs, procedures, rules, and any associated documentation pertaining to the operation of a system.

Software Configuration Index (SCI)

`Certification - SUITE/DISPLAY`

A document that identifies the configuration of a software product.

Software life cycle

`Certification - SUITE/DISPLAY`

Activities occurring during a period of time that starts when software is conceived and ends when the software is no longer available for use. The software life cycle typically includes a requirements phase, development phase, test phase, integration phase, installation phase, and a maintenance phase.

Software verification tools

`Verification - SUITE/DISPLAY`

Tools used to verify the code (*e.g.,* simulators, test execution tool, coverage tools, reporting tools). Verification tools cannot introduce errors, but may fail to detect them. For example a static analyzer that automates a software verification process activity, should be qualified if the function that it performs is not verified by another activity. Type checkers, analysis tools and test tools are other examples.

Source Code Control

`Management - SUITE/DISPLAY`

Source Code Control API defined by Microsoft as a standardized access means to Configuration Management Systems.

Specialization operator

`Design - SUITE`

An operator that defines a specific behavior for a set of actual data types from the instances of a specialized operator. See Specialized operator.

Specialized operator

`Design - SUITE`

A polymorphic operator declared as imported in the model. Its behavior is defined by one or several specialization operators. Instantiating a specialized operator in models computes the correct specialization to set its specific behavior. See Polymorphic operator and Specialization operator.

Specification

`Design - DISPLAY`

A SCADE Display model which corresponds to an SGFX file. It is composed of at least one layer and is characterized by its size in pixels or user units. It may be a complete image description or only a small part of it.

SST files

`Design - DISPLAY`

A file in which is stored the specification line stipple table.

Standalone executable

`Verification - SUITE`

An executable embedding both a model and a graphical panel, and that runs independently on any Windows/PC (no run-time fee). The standalone executable enables managing the model's inputs/outputs interactively from the graphical panel.

Standard

`Verification - SUITE`

A rule or basis of comparison used to provide both guidance in and assessment of the performance of a given activity or the content of a specified data item.

State

`Design - SUITE`

The basic memory element of a state machine. At each cycle, a state is either active or inactive.

State machine

`Design - SUITE`

See SCADE State Machine.

Static expression

`Design - SUITE`

See Constant expression.

Step-by-step simulation/animation

`Verification - SUITE/DISPLAY`

A simulation mode that allows entering input values on determined model inputs and at specific simulation cycles.

Stop condition

`Design - SUITE`

An advanced breakpoint. A stop condition is used if a more complex expression is needed for a breakpoint.

Strong preemption

`Design - SUITE`

A state machine transition that deactivates the source state and activates the target state to start its internal activity.

Structure tree

`Design - DISPLAY/RAPIDPROTO`

The structure of the model displaying all the specified objects, hierarchically ordered.

Surface-based primitive

`Design - DISPLAY/RAPIDPROTO`

A primitive based on a closed contour done of a set of segments and arcs. It is done of both the contour (outline) itself and of the associated surface (filling).

SWT files

`Design - DISPLAY/RAPIDPROTO`

A file in which is stored the specification line width table.

Symbol files (SSL)

`Design - SUITE`

Files in which are saved the graphical representation associated with any SCADE operator.

Symbol definition

`Design - ARINC661`

A file containing a tree of Symbol Definition commands. Each Symbol definition command has field parameters, also named “command properties”. The type of the command defines by reference to the A661 Configuration Data the expected command properties.

Symbol table

`Design - ARINC661`

A table containing a list of symbol Ids associated with Symbol Definition files. Several ARINC 661 widgets (*e.g.*, Symbol widget) and map items (SYMBOL_GENERIC, SYMBOL_ROTATED and SYMBOL_TARGET) refer to symbols by a Symbol ID number.

Symbology container

`Design - DISPLAY/RAPIDPROTO`

Object used to structure the images in the layer. It allows grouping objects together and associating a dynamic behavior to these groups.

Symbology layer

`Design - DISPLAY`

See Layer.

Synchro preemption

`Design - SUITE`

A state machine transition that is triggered when a state terminates (*e.g.*, all its sub-state machines are in terminal state at the end of the cycle) and starts the activity of the target state on next cycle.

Synchronous mode

`Verification - SUITE`

An integration mode for co-simulation between behavioral logic and graphical panel components. In this mode, the graphical panel runs synchronously with behavioral logic model (*e.g.*, when simulation is paused, graphical panel is inactive). See Asynchronous mode.

Syntactic check

`Verification - SUITE`

SCADE Suite Code Generator performs a lexical and syntactical analysis of the description of the input model. Its scope of analysis is the file content as well as all files included in the SCADE Suite project.

SysML/UML models

`Design - SUITE`

See Architecture Design.

T

Test Automation Framework

`Verification - ARINC661`

A toolset providing comprehensive services for the testing automation of ARINC 661 widgets.

Test case

`Verification - SUITE/DISPLAY`

A set of test inputs, execution conditions and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement.

Test Environment for Host

`Testing - TEST/SUITE/DISPLAY`

SCADE Test Environment for Host provides an interactive and a batch user interface for applications developed in SCADE Suite and SCADE Display. From the interactive interface, test engineers can create and manage test data, setup and launch test execution, and get summarized and detailed test execution reports.

Test Execution batch mode on host

`Verification - TEST`

The batch mode is a certified chain dedicated to execute tests and produce conformance report for certification activities. SCADE Test Environment is qualified as a verification tool for DO-178C certification and only this batch mode can be used for certification purposes.

Test harness

`Verification - TEST`

A test harness is a piece of software and/or hardware calling the component to test and making test easier.

Test procedure

`Verification - TEST`

Detailed instructions for the set-up and execution of a given set of test cases and instructions for the evaluation of results of executing the test cases.

Test result project

`Testing - TEST`

A test result project created for storing and keeping the results from various testing campaigns. A test result project references a test project.

Test scenario

`Verification - TEST`

Scenarios that can be run in SCADE Test tools to evaluate the execution of test data on models or measure their contribution to the coverage of the model.

Test Target Execution

`Testing - TEST`

SCADE Test Target Execution automates the generation of target test harness for COTS tools by translating model test cases into test harnesses.

Text-based primitive

`Design - DISPLAY/RAPIDPROTO`

Primitive based on an associated font, where a font describes the shape of each character.

Textual diagram

`Design - SUITE`

Diagrams that contain definitions in Scade textual language. All of the textual diagram content is editable.

Texture

`Design - DISPLAY/RAPIDPROTO`

A graphic pattern that can be applied on a surface-based object to imitate surface shading, lighting, and color. The texture is described by RGB or RGBA components which are called texel. Texture mapping is required to apply a texture to a graphic primitive (the texture map is a file containing the desired image): for each vertex of the primitive, the corresponding mapping on the texture has to be specified.

Timing and Stack Optimizer (TSO)

`Verification - SUITE`

Tool that enables to evaluate the worst-case execution time (WCET) and stack size estimation for different versions of a SCADE Suite model to determine the most efficient design. It allows comparing analysis session results and also visualizing them graphically with AbsInt's aiSee.

Timing and Stack Verifiers (TSV)

`Verification - SUITE`

Tools that enable to compute the worst-case execution time (WCET) and stack usage for a SCADE Suite model with respect to specific target processors and C compilers. Tools allow comparing analysis session results and also visualizing them graphically with AbsInt's aiSee.

Tool Accomplishment Summary (TAS)

`Certification - SUITE/DISPLAY`

Presents the compliance status with the Tool Qualification Plan, the usage conditions, the list of unresolved bugs, and the possible limitations of the tool.

Tool certification

`Certification - SUITE/DISPLAY`

The process necessary to obtain certification credit for a software tool within the context of a specific system.

Tool Operational Requirements (TOR)

`Certification - SUITE/DISPLAY`

A document that describes the functions and technical features of the tool, the tool operational environment and user information. This document is not exhaustive on the tool behavior but presents the functions to be qualified.

Tool Qualification Data

`Certification - SUITE/DISPLAY`

Set of data obtained after the Tool Qualification activities performed for qualification of KCG code generators. The KCG development qualification data are available for audit, including design and source code, verification cases and procedures, verification results, software configuration management and software quality assurance records.

Tool Qualification Plan (TQP)

`Certification - SUITE/DISPLAY`

A document that describes the tool qualification strategy and processes and identifies the tools, personnel and competence needed to achieve the tool qualification.

Tool qualification/certification

`Certification - SUITE/DISPLAY`

The process necessary to obtain certification credit for a software tool within the context of embedded safety critical systems.

Traceability

`Management - SUITE/DISPLAY`

The evidence of an association between items, such as between process outputs, between an output and its originating process, or between a requirement and its implementation.

Transformation

`Design - DISPLAY/RAPIDPROTO`

Pre-processing of an object inputs performed in a functional container (translation or rotation container) to allowing to reduce cost of a graphic change of that object.

Transitions

`Design - SUITE`

State machine components that connect a source state to a target state. Whenever the source state is active, the transition preempts the activity of the source state and starts the activity of the target state.

Triangle strip

`Generation - DISPLAY/RAPIDPROTO`

A shape drawing mode used to draw triangles from vertices: the vertices are taken 3 by 3 to draw a triangle: a first triangle is drawn with the first, the second, and the third vertex. A second triangle is drawn with the second, the third, and the fourth vertex. A third triangle is drawn with the third, the fourth, and the fifth vertex, etc.

TrueType

`Design - DISPLAY/RAPIDPROTO`

An outline (or vector) font standard developed by Apple and shared with MicroSoft.

Type

`Design - SUITE`

See Data typing.

Type propagation

`Design - SUITE`

When creating multiple connections consecutively, SCADE Suite Editor propagates types automatically along connecting wires in the whole design provided all typing information is recovered.

U

UA

`Design - ARINC661`

The acronym for User Application.

Underflow

`Verification - SUITE`

A situation regarding real number representation with float number happening when the minimal positive value closest to 0 (zero) representable by a data storage is greater, in absolute value, than the value it is supposed to store. For example, a 32bits IEEE 754, 10e-100 induces an underflow as it represented as 0 (zero).

Unified Modeling Language (UML)

`Design - SUITE`

A standardized general-purpose modeling language. UML includes a set of graphical notation techniques to create abstract models of specific systems, referred to as UML model.

User Application

`Design - ARINC661`

A software component connected with the CDS, using the ARINC 661 protocol to exchange information.

User coordinates

`Generation - DISPLAY`

Coordinates expressed in the ratio of the specification.

User-defined operator

`Design - SUITE`

User-defined operators are built from a combination of predefined operators and/or other user-defined operators. They can be put into libraries. Any operator created in SCADE Suite model is referred to as a user-defined operator to distinguish them from the predefined operators defined in Scade language. See Library.

V

Validation

`Verification - SUITE/DISPLAY`

Activity of demonstration, by analysis and test, that the product meets, in all respects, its specified requirements.

Variables Dictionary

`Design - DISPLAY/RAPIDPROTO`

A dictionary allowing for the management of variables in a specification.

Verification

`Verification - SUITE/DISPLAY`

Activity of determination, by analysis and test, that the output of each phase of the life cycle fulfils the requirements of the previous phase.

Verification strategy

`Verification - SUITE`

In formal verification activities, it is possible to specify which verification strategies to use when analyzing proof objectives, to run property analysis. The strategies available for verification are:

- Debug strategy to track errors and bugs in the design over a certain amount of execution cycles to be analyzed. Considering safety requirement properties, if the design violates a given property, this strategy finds a counter-example scenario very fast.
- Prove strategy to prove that a property is valid. As soon as a property is falsifiable, this strategy generates a counter-example like the Debug strategy does. This is the strategy used as default strategy for new proof objectives.

Viewport

`Generation - DISPLAY`

The defined area on the screen, which limits the pixels available for object rendering.

Visibility

`Design - DISPLAY/RAPIDPROTO`

An attribute that defines the visibility of an object. It allows not displaying the object.

W

WCET

`Verification - SUITE`

The acronym of Worst Case Execution Time.

Weak preemption

`Design - SUITE`

A state machine transition that allows the source state to finish its activity and delays the start of the target state activity till next cycle.

WGFX file

`Design - RAPIDPROTO`

A file describing a part of a graphical panel and that can be used as a widget.

White-box simulation

`Verification - SUITE`

Simulation of (a piece of) a SCADE Suite model based on the observation of its internal behavior. The graphical access to all SCADE Suite diagrams and variables allows at the same time analyzing and debugging the outputs.

Widget

`Design - DISPLAY/ARINC661/RAPIDPROTO`

An individual element of the GUI. Each widget belongs to a widget type which is described in the ARINC 661 standard.

Widget Extension

`Design - ARINC661`

A mechanism to define optional properties for some widget types. The allowed extension properties are defined in a Widget Extension file in the A661 Configuration Data. The list of authorized widget extension types, for each widget type is defined in the Widget Extension Hierarchy file.

Widget Libraries

`Design - DISPLAY/ARINC661/RAPIDPROTO`

Two sets of libraries:

- A first set of predefined widgets (such as buttons, knobs, sliders, LEDs, etc.) accessible from SCADE Rapid Prototyper and used to create graphical panels for running interactive simulation sessions, or generating standalone executables.
- A second set of SCADE “sources” (models, C code, XML configuration files) corresponding to the design of the ARINC 661 widgets listed in with the ARINC 661-4 specification.

Wires

`Design - SUITE`

Links used to connect design element in SCADE Suite projects.

Worst Case Execution Time

`Verification - SUITE`

A task that consists of analyzing the maximum length of time the task could take to execute on a specific hardware platform.

See also Timing and Stack Verifiers and Timing and Stack Optimizers.

X

XML

`Design - SUITE`

The acronym of eXtended Markup Language.

XML Configuration file

`Design - ARINC661`

An XML file that contains the data required to define the widgets contained in the customer widget library (list of widgets, definition of types and constants, hierarchy, interfaces, implementation). This file is editable in UI.

XSCADE files

`Design - SUITE`

Files that contain design functions in XML format. All elements created in SCADE Suite Editor are stored in this format.

Contact Information

Submit questions to Technical Support at
scade-support@ansys.com

Contact one of our Sales representatives at
scade-sales@ansys.com

Direct general questions about SCADE products to
scade-info@ansys.com

Discover the latest news on our products at
www.ansys.com/products/embedded-software

Revision: SCF-GLO-25 - 04/10/24