What's New in Version 1.0?

Find out what's new in the version 1.0 release of SCCharts!

What is new in **SCCharts** Version 1.0?

One main change is the new textual SCCharts syntax using the new `.sctx` file extension. See Converting Legacy Models (sct) on how to convert your existing models.

Controlflow / Dataflow Hybrid-Models

- You can now add dataflow regions to your SCCharts models (see Syntax#Dataflow)

Semantic Comments

- You can now add semantic comments to your SCCharts models (see Syntax#Comments)
- You can also alter the style of the comments to mark POI in your program

New Reference / Module Support / Experimental Inheritance

- The reference and module support for SCCharts has been reworked (see Syntax#References)
- The content assist will help the modeler with the model bindings
- Multiple SCCharts models can be stored in one file
- SCCharts now supports experimental inheritance

For Region

- Regions can be duplicated automatically (see Syntax#ForRegions)
- For regions have access a unique iterator variable

Vectors

- You can now use vector values to assign arrays. (see Syntax#Vectors)
- Vectors can also be used in the dataflow models.

Further Experimental Features

- Added Scheduling Directives
- Added Probabilistic Transitions
- Added timed automata support for SCCharts

New Expressions

- Expression language now supports infix assignment operators
- Expression language now supports bitwise xor and bitwise not
- Expression language now supports shift operators
- Expression language now supports the ternary conditional operator
- Operator precedences are now correctly mapped to the kexpressions model structure
- Improved host code expressions support

New Annotations & Pragmas

- Added pragma support to differ between model element annotations and file specific pragmas
- Added unicode pragmas
- Added compiler pragmas
- Added generic layout annotations

SCCharts Syntax Changes

- Primes can be used in identifier
- Hostcode now uses accent grave (`)
- Function call syntax with angle brackets is considered deprecated. It can still be accessed with the **extern** keyword
- Semicolon now exclusively stands for the sequence operator. By default you don't need a semicolon as line/command delimiter.
- To see a complete overview of the SCCharts syntax, please consult our SCCharts syntax page.

What is new in **KIELER** SCCharts Version 1.0?

New Compiler Framework (a.k.a. KiCo 3.0 a.k.a. KiCool)

- Compilation systems are now models which can be configured and instantiated at run-time
- Added pre-configured systems for all existing compilation approaches and tests
- Redone all associated views
    - Added multi-select in side-by-side mode
- Integrated model element tracing
- Integrated valued object meta information

New Code Generation for SCCharts

- The code generation now creates functions that can handle different status instances of a program.
- The model status is generally saved to a dedicated struct.
- Added new state-based code generation approach for C
- Added new lean state-based code generation approach for C
- There now exist modular compilation systems for all supported compilation approaches
    - Netlist-based C
    - Netlist-based Java
    - Priority-based C
    - Priority-based Java
    - State-based C
    - Lean State-based C
- All compilation systems have additional variants for simulation and tests

New Supplementary Code Generations

- Original Esterel
- SCEst
- Experimental compilation to Lustre
- Experimental compilation from Lustre

New Processors

- Besides the mandatory processors for the a.m. code generators there are several optional new processors for academic and experimental purposes
    - generic SSA
    - Loop analyses and compiler optimizations (e.g. copy & constant propagation)
    - External compiler invocation
    - Arduino deployment
    - Eclipse project setups
    - Structural Depth Join (SDJ) for schizophrenic models

New Warnings & Errors

- Detailed reference warnings
- Label shadowing

Integrated Simulation

The simulation backend has been rewritten to be more lightweight, flexible and transparent and to better integrate in the workflow.

Major new features are:

- Improved usability for simulations (one-click simulations)
- Support for arrays and internal variables, such as SCG guards
- Full integration of simulation code generation into KiCo and the project structure (KIELER-Temp project)

Simulation Visualization

- Added dedication data view for simulation values
- Added live values inside the model diagram
- Simulation visualization view that links an SVG image to the program state, using a mapping and javascript commands defined in a kiviz file.

What is new in KIELER SCCharts Version 1.0 for **Developers**?

Grammar Changes

- A dash (-) is now available in ExtendedIDs. SCCharts States may now include dashes in their IDs.
- Single underscore (_) IDs are no longer valid. Underscores prefix generated IDs or are *Value* keywords.

KExpressions

- Valued Objects in assignments and emissions are now valued object references. Hence, they reuse the KExpressions concepts.
    - If you `@inject KEffectsExtensions`, you can use the same syntax as before. Otherwise, `assignment.reference` is the `ValuedObjectReference` that points to the referenced `ValuedObject. You can use the reference as usual. Keep in mind that it is a containment.`

`Workflow`

- The compiler framework now uses KiCo 3.0. Please consult the developer documentation of KiCo for further questions.
- The simulation framework now uses the V3 simulation based on KiCo 3.0. Please consult the developer documentation of the simulation for further questions.
- Most of the test cases now uses the models repository directly. You can specify the location of you models repository in the `models_repository` variable in your test launch configuration. Please consult the developer documentation of the test framework for further questions.