Accellera SystemC Working Groups Update

SystemC Evolution Day 2024

Martin Barnasconi

Accellera Technical Committee Chair

accellera.org

SYSTEMC EVOLUTION DAY
OCT 17, 2024 | MUNICH | GERMANY

Outline

- Accellera Systems Initiative & Working Groups
- SystemC ecosystem
- SystemC Working Groups update
- Federated Simulation Working Group and User Group
- systemc.org
- How to join us

Accellera Systems Initiative

Our Mission

To provide a platform in which the electronics industry can collaborate to innovate and deliver global standards that improve design and verification productivity for electronics products.

PORTABLE STIMULUS
SYSTEM C™
SystemVerilog
VHDL
UVM
IP-XACT™
SystemRDL™
UPF UNIFIED POWER FORMAT

Accellera Working Groups

Verification-centric Working Groups

- Portable Stimulus
- Multi-Language
- SV-AMS
- UVM
- UVM-AMS

SystemC Working Groups

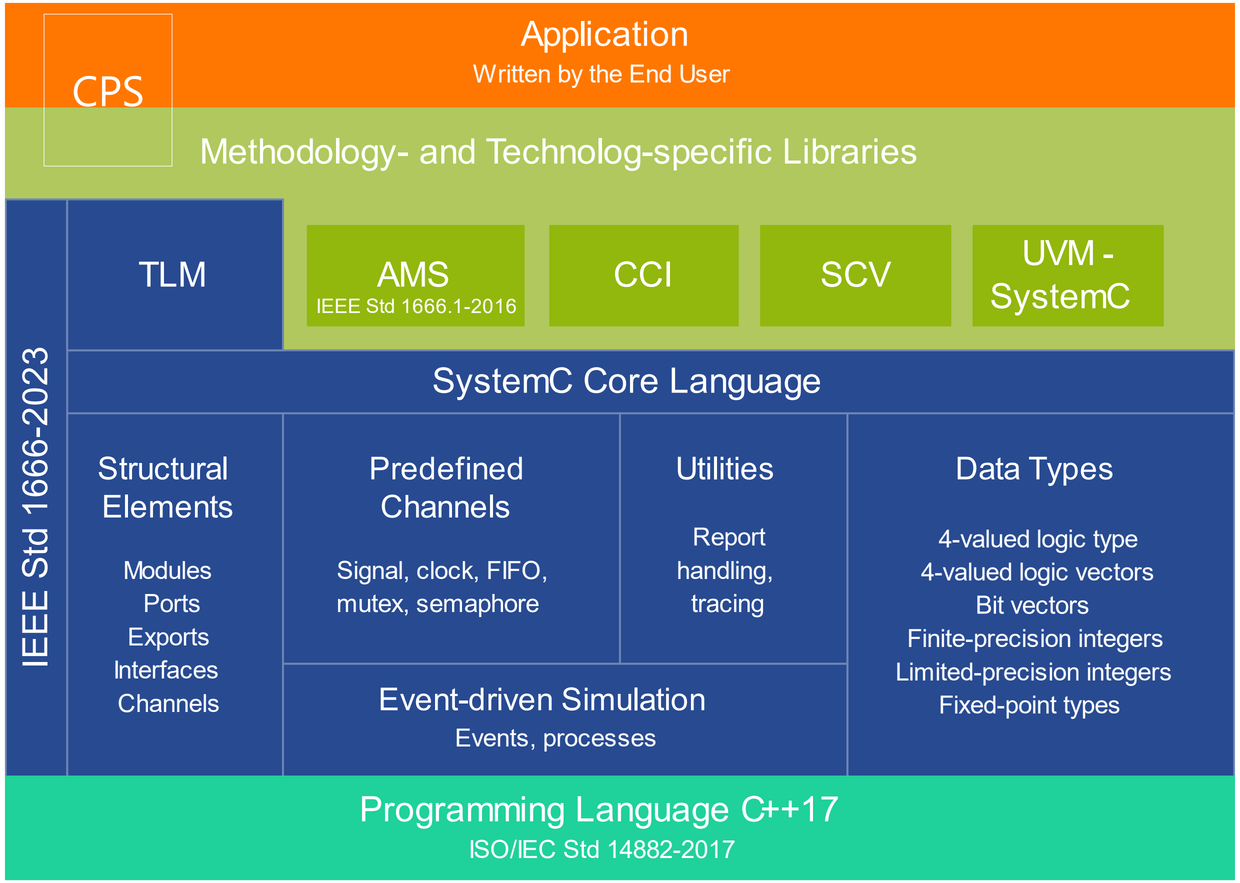
- Language
  - DT
  - CPS
- AMS
- CCI
- Verification
- Synthesis

Working Groups in other domains

- Functional Safety
- IP-XACT
- CDC / RDC
- FSS (+UG) NEW

SystemC ecosystem

- SystemC is a C++-based language standard, widely used for
  - System-level modeling, design and verification
  - Architectural exploration, performance modeling
  - Analog/mixed signal modeling
  - High-level Synthesis
  - Software development
- Released as IEEE standards
  - IEEE Std. 1666-2023 (SystemC)
  - IEEE Std. 1666.1-2016 (SystemC AMS)



CPS

Application
Written by the End User

Methodology- and Technolog-specific Libraries

IEEE Std 1666-2023

TLM

AMS
IEEE Std 1666.1-2016

CCI

SCV

UVM - SystemC

SystemC Core Language

Structural Elements
Modules
Ports
Exports
Interfaces
Channels

Predefined Channels
Signal, clock, FIFO, mutex, semaphore

Utilities
Report handling, tracing

Data Types
4-valued logic type
4-valued logic vectors
Bit vectors
Finite-precision integers
Limited-precision integers
Fixed-point types

Event-driven Simulation
Events, processes

Programming Language C++17
ISO/IEC Std 14882-2017

More information: https://systemc.org/

SYSTEMC™ EVOLUTION DAY
OCT 17, 2024 | MUNICH | GERMANY

Accellera SystemC Working Groups

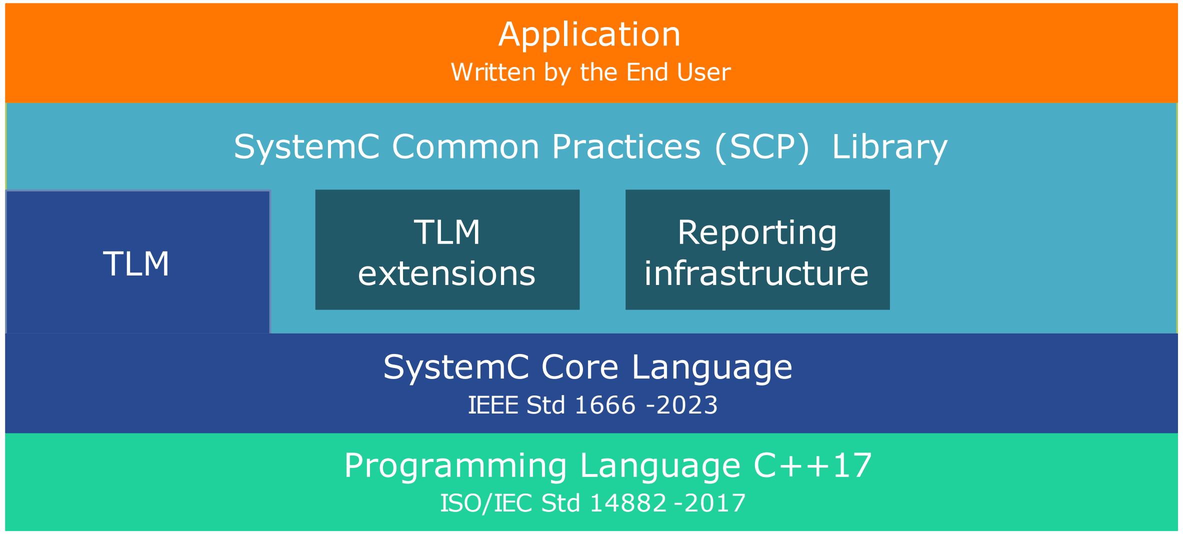
- SystemC Language Working Group (LWG)
  - Chair: Laurent Maillet-Contoz (ST)
  - Subgroup: Common Practices (SCP): Chair: Mark Burton (Qualcomm)
- SystemC Analog/Mixed-Signal Working Group (AMSWG)
  - Chair: Martin Barnasconi (NXP)
- SystemC Configuration, Control & Inspection Working Group (CCIWG)
  - Chair: Lukas Jünger (MachineWare)
- SystemC Synthesis Working Group (SWG)
  - Chair: **Frederic Doucet (Qualcomm)**
- SystemC Verification Working Group (VWG)
  - Chair: Stephan Gerth (Bosch)

SystemC Language Working Group

- SystemC 3.0.1 update to be published soon
  - Made available via Accellera public repository on GitHub: https://github.com/accellera-official/systemc/tags
  - Includes improvements CMake flow, address bugfixes in data-types, regression tests, etc.
- Next steps
  - CMake will become the default build system (legacy Automake flow kept as fallback for now)
  - Continue to collect input and requirements for next standardization cycle: next revision of IEEE 1666 and reference implementation SystemC 4.0
- Encouraging to see more and more issues reported by community members
  - Complementing issue report with test case and/or pull request would help the LWG to assess the issue
- IEEE SA P1666 WG finalizing Corrigendum as part of the IEEE 1666 SystemC LRM

SystemC Common Practices Working Group

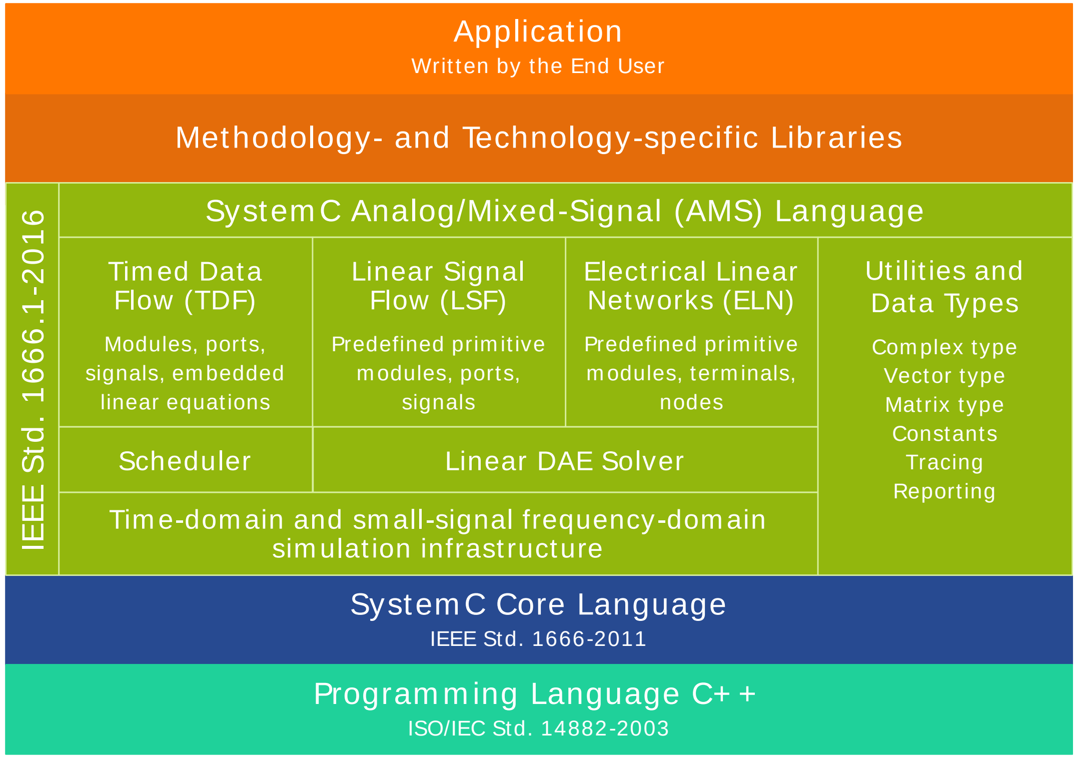
- Proposal to improve reporting and logging as part of SystemC core language
- Prototype implementation available in public repository
  - https://github.com/accellera-official/systemc-common-practices
- Improvements under consideration for next standardization round (to be agreed/aligned with SystemC LWG)



More information: https://systemc.org/overview/systemc-scp/

SystemC Analog/Mixed-Signal (AMS) WG

- Documentation of Language Reference Manual (LRM) for next IEEE 1666.1 update started, targeting
  - Piece-wise-linear support for ELN and LSF primitives
  - Converter primitives between LSF and ELN MoC
  - Solver API
  - Threshold detection
- Call for participation to IEEE SA P1666.1 WG early 2025
  - Target for finalization: 2026



Application
Written by the End User

Methodology- and Technology-specific Libraries

IEEE Std. 1666.1-2016

SystemC Analog/Mixed-Signal (AMS) Language

Timed Data Flow (TDF)
Modules, ports, signals, embedded linear equations

Linear Signal Flow (LSF)
Predefined primitive modules, ports, signals

Electrical Linear Networks (ELN)
Predefined primitive modules, terminals, nodes

Utilities and Data Types
Complex type
Vector type
Matrix type
Constants
Tracing
Reporting

Scheduler

Linear DAE Solver

Time-domain and small-signal frequency-domain simulation infrastructure

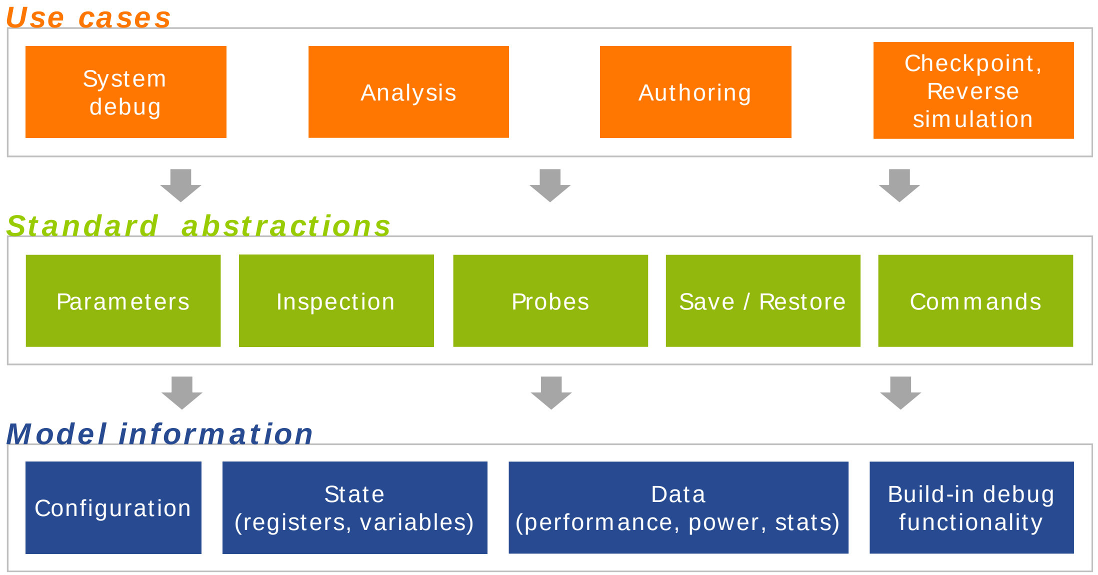
SystemC Core Language
IEEE Std. 1666-2011

Programming Language C++
ISO/IEC Std. 14882-2003

More information
https://systemc.org/overview/systemc-ams/

SystemC Configuration, Control & Inspection WG

- CCI reference implementation including latest developments is publicly available
  - https://github.com/accellera-official/cci/
- Established regression/CI flow in GitHub
  - Other WGs will follow
- Proposal and demonstrator available for Inspection API
  - More details in the other presentation today!
- Other areas of discussions/alignment
  - Control/command API
  - Common strategy for object/name look-up



Use cases

System debug | Analysis | Authoring | Checkpoint, Reverse simulation

Standard abstractions

Parameters | Inspection | Probes | Save / Restore | Commands

Model information

Configuration | State (registers, variables) | Data (performance, power, stats) | Build-in debug functionality

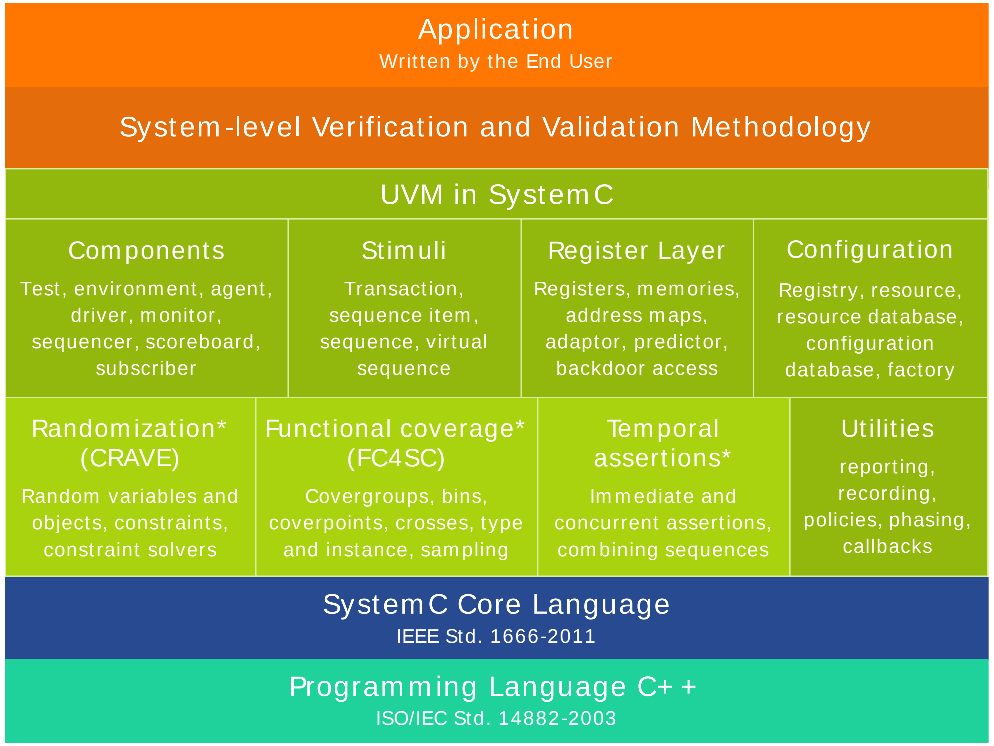
More information
https://systemc.org/overview/systemc-cci/

SystemC Synthesis WG

- SystemC Synthesis Working Group restarted early 2024
  - New leadership: Chair Frederic Doucet (Qualcomm), Vice-chair Rauf Salimi Khaligh (Apple)
  - Call for Participation: https://vimeo.com/929513471
- WG plans and developments presented at SystemC Fika in May
- Update of SystemC Synthesizable Subset ongoing (v1.5)
- Discussion topics / Roadmap items
  - Unambiguous interpretation SystemC syntax for synthesis
  - HLS Constraints and Directives;
  - Communication interfaces (e.g., channels, memories, etc.)
  - Modern language constructs (e.g., C++17/20)
  - Synthesizable Data types
  - SoC/infrastructure libraries and common design building blocks
  - Proof-of-concept implementation and examples

SystemC Verification Working Group

- UVM-SystemC library 1.0beta6 was released in July 2024
  - https://www.accellera.org/images/downloads/drafts-review/uvm-systemc-1.0-beta6.tar.gz
  - Compatible with SystemC 3.0.0
- Documentation of UVM-SystemC class library in Language Reference Manual (LRM)
- Ongoing alignment with UVM (SystemVerilog) Working Group
  - UVM-SV TLM vs SystemC TLM
  - Standardized API vs exposed methods offered in the reference implementation



Application
Written by the End User

System-level Verification and Validation Methodology

UVM in SystemC

Components
Test, environment, agent, driver, monitor, sequencer, scoreboard, subscriber

Stimuli
Transaction, sequence item, sequence, virtual sequence

Register Layer
Registers, memories, address maps, adaptor, predictor, backdoor access

Configuration
Registry, resource, resource database, configuration database, factory

Randomization* (CRAVE)
Random variables and objects, constraints, constraint solvers

Functional coverage* (FC4SC)
Covergroups, bins, coverpoints, crosses, type and instance, sampling

Temporal assertions*
Immediate and concurrent assertions, combining sequences

Utilities
reporting, recording, policies, phasing, callbacks

SystemC Core Language
IEEE Std. 1666-2011

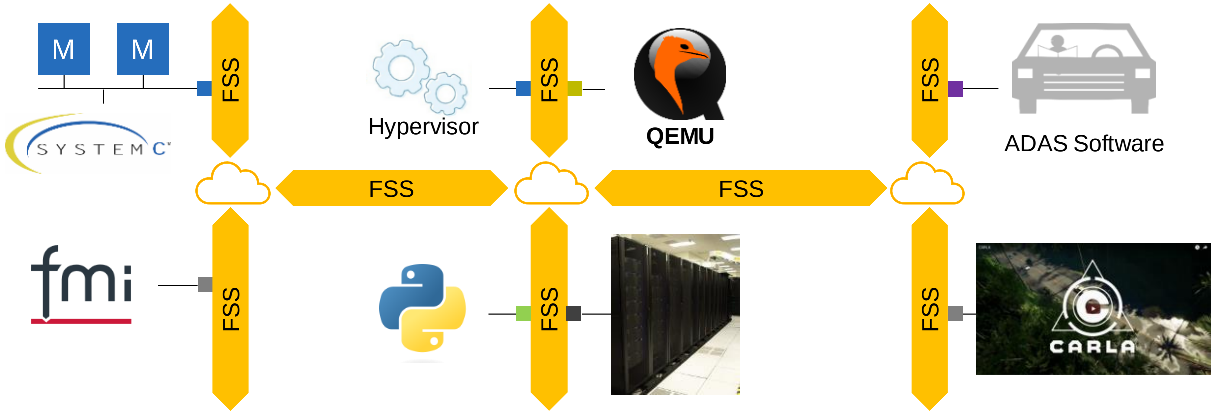
Programming Language C++
ISO/IEC Std. 14882-2003

* Integration on Roadmap

More information
https://systemc.org/overview/systemc-verification/

Federated Simulation Working Group and User Group

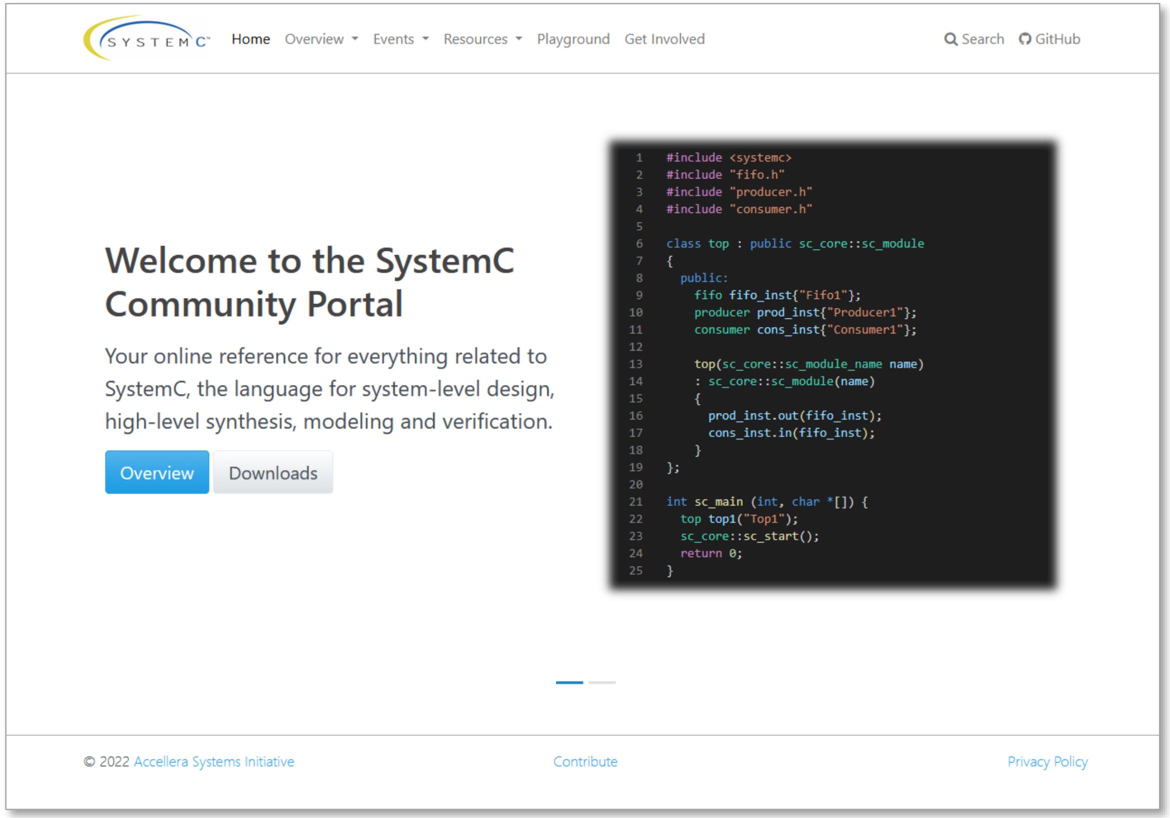
- Federated Simulation Working Group and User Group established in September 2024
- Objective: Introduce standardized interfaces
  - Enabling interoperability between simulation frameworks
- Targeting a scalable simulation and modeling ecosystem
  - Support models and simulation domains used at different levels of the ‘OSI stack’
  - Ecosystem of simulators, models, and other components that together form Systems-of-Systems
- User Group is open for the community!
  - Accellera members are welcome to join the working group



Automotive example

systemc.org

- **The** SystemC community portal, containing
  - SystemC overview pages covering all Working Groups
  - SystemC Evolution Day Events and Fikas: all presentations and videos
  - Open Access Publications
  - Libraries and Projects
- **YOU** can help in adding content!
  - Submit your pull request to github.com/accellera-official/systemc.org



Welcome to the SystemC Community Portal

Your online reference for everything related to SystemC, the language for system-level design, high-level synthesis, modeling and verification.

```
#include <systemc>
#include "fifo.h"
#include "producer.h"
#include "consumer.h"

class top : public sc_core::sc_module
{
  public:
    fifo fifo_inst{"Fifo1"};
    producer prod_inst{"Producer1"};
    consumer cons_inst{"Consumer1"};

    top(sc_core::sc_module_name name)
    : sc_core::sc_module(name)
    {
      prod_inst.out(fifo_inst);
      cons_inst.in(fifo_inst);
    }
};

int sc_main (int, char *[]) {
  top top1("Top1");
  sc_core::sc_start();
  return 0;
}
```

© 2022 Accellera Systems Initiative

How to join us

- Become an Accellera Working Group member
  - Join Accellera and participate in the Accellera working groups
  - Direct access to the latest standardization proposals and development implementations
- Become a member of the IEEE Standards Association
  - Join IEEE-SA to participate in the IEEE P1666 (SystemC) working group
- Share your experiences
  - Visit www.accellera.org and join the community forums at forums.accellera.org
  - Report your issues and/or create pull requests on the public SystemC GitHub repository
- Help us to grow the SystemC ecosystem and community
  - Participate in community events such as the SystemC Evolution Day and Fika
  - Contribute to the SystemC Community Portal systemc.org
  - Promote the use of the SystemC standard in complex system simulation tasks

Thank You

Q&A