

# SystemC 2.1 Preview - DAC2004



Electronic Systems Solutions

Presented by David C. Black, co-founder

[www.EklecticAlly.com](http://www.EklecticAlly.com)

[info@EklecticAlly.com](mailto:info@EklecticAlly.com)

Voice mailbox/FAX 888-467-4609

version 1.0

# Motivations

---

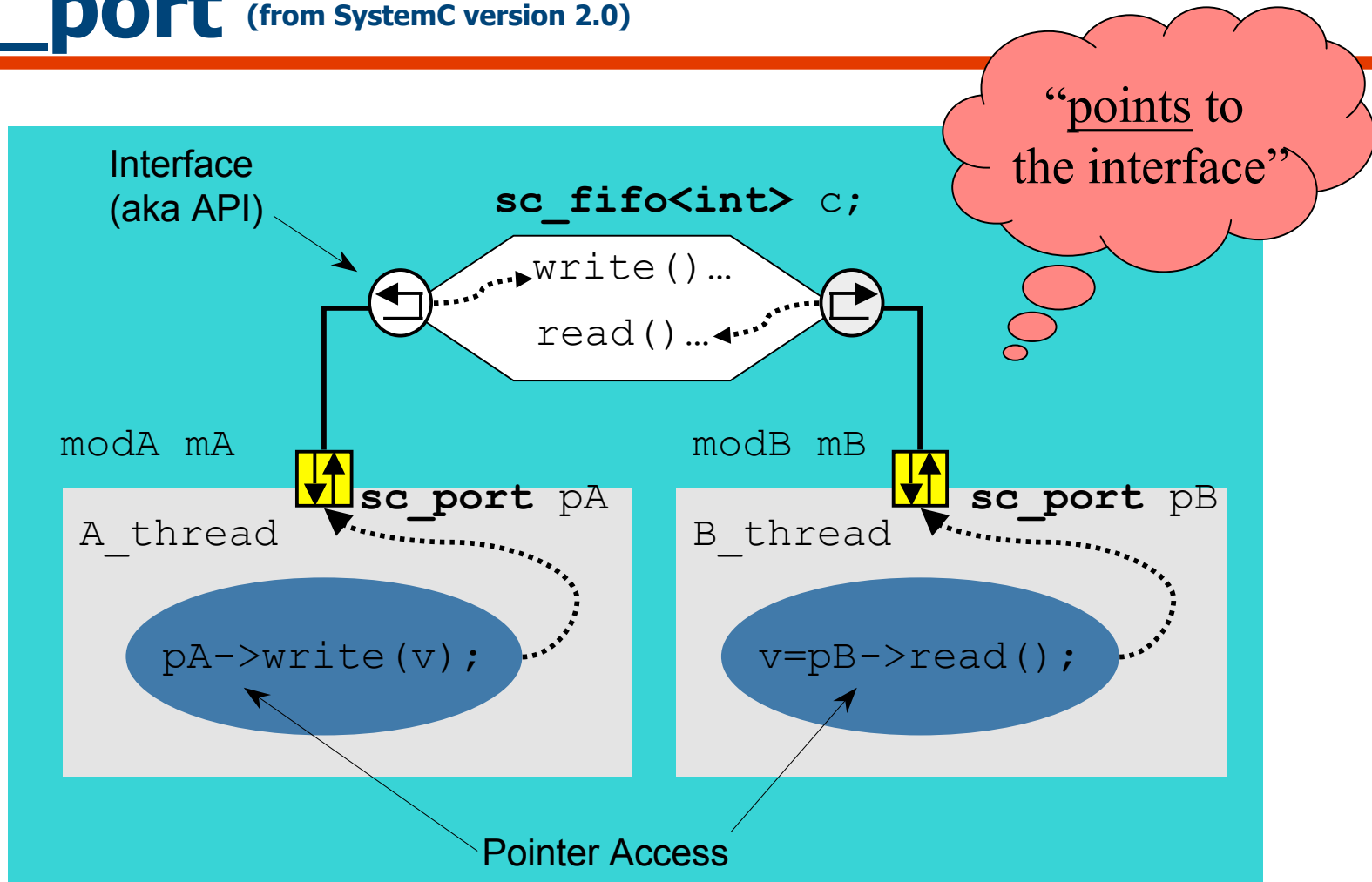
- Improve modeling features
- Improve verification features
- Improve modularity for IP
- Ease of use
- Bug fixes

# Improved modeling features

---

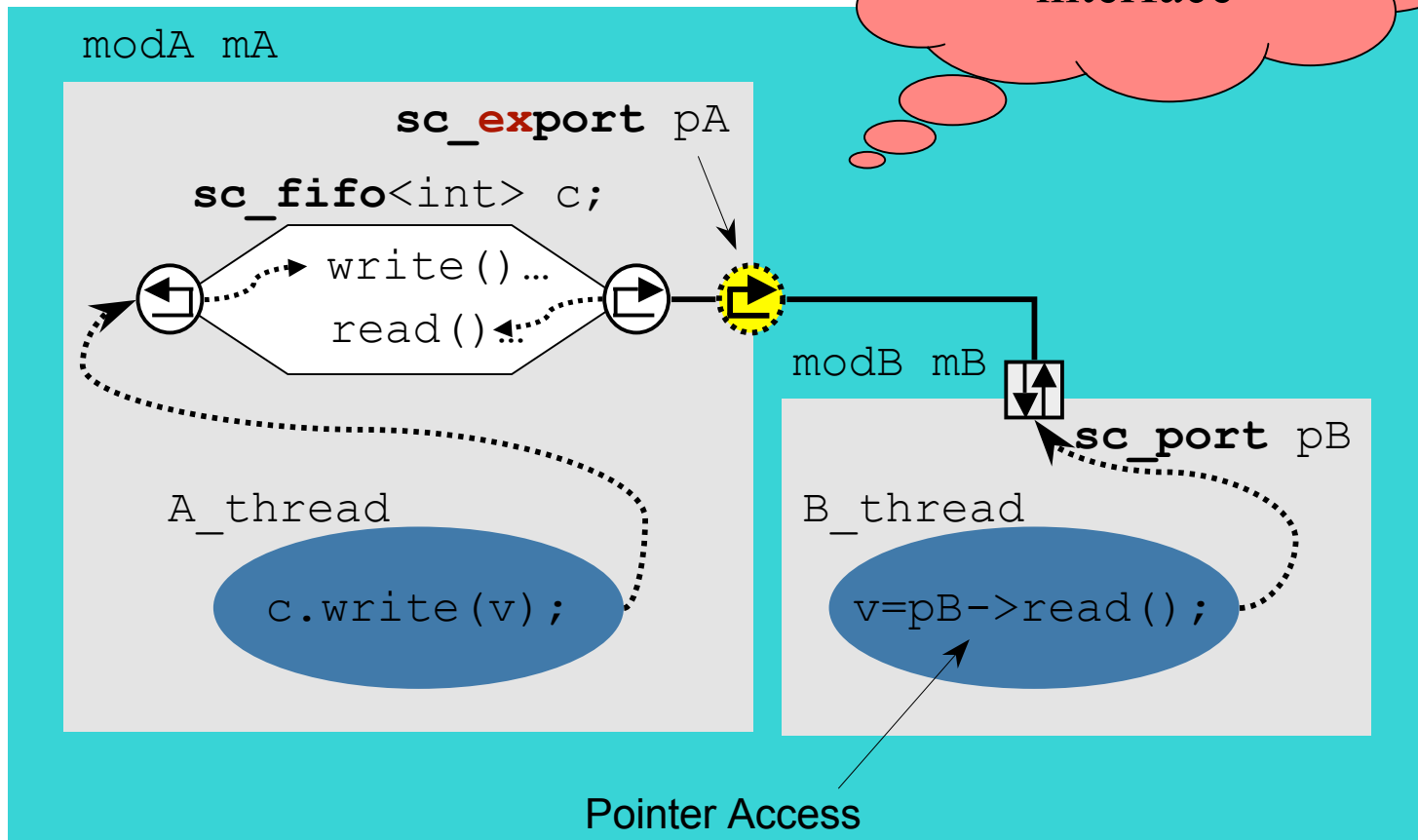
- `sc_event_queue` class
  - Multiple notifications result in multiple events
  - Reliably catch every notification
- `sc_export`
  - Opposite of `sc_port` w.r.t. caller/callee
  - Exposes internal channels
  - More controlled hierarchical interfaces
  - Can reduce context switching

# sc\_port (from SystemC version 2.0)



# sc\_export (new for SystemC version 2.1)

“exports the interface”



# Improved verification features

---

- Dynamically spawned processes
  - Fork/join
  - Required for temporal assertion checking
  - Helpful for testbenches
  - Automatic allocation/deallocation of threads
  - Multiple threads off a single method

# Improved modularity for IP

---

- `sc_export` - previously mentioned
- Structured error reporting
  - Consistent messaging for all components
    - ◆ Simulator core, libraries, IP, modules, testbenches
- Easier access to startup arguments
  - Simplifies ability of libraries and IP to use command-line control
  - `sc_argc()` and `sc_argv()`
- New callbacks allow IP integration with requiring code in `sc_main`
  - `before_end_of_elaboration()`
  - `start_of_simulation()`
  - `end_of_simulation()`

# Structured error reporting

---

my\_module.cpp

```
extern char* sim_nm;
void my_module::some_thread() {
    SC_REPORT_INFO(sim_nm, "Starting");
    ...
    if (error_condition) {
        SC_REPORT_ERROR(sim_nm,
            "Oops something bad happened");
    }
    ...
}
```



# Output from reporting

---

```
0 ns: Info: MY_SIM: Starting
...
4 ns: Error: MY_SIM: Oops something bad happened
In file: my_module.cpp:24
In process: my_mod_i.some_thread_0 @ 4 ns
```

# Ease of use

---

- Signal and port specializations
  - Assign to/from part selects of signals and ports directly
- Mixed concatenation
  - Concatenations of `sc_(u)int/sc_big_(u)int` can now be mixed without ugly casting
- Object code release tagging
  - Link-time detection between incompatible object files
- POSIX thread support
  - Allows use of code coverage and memory leak checking tools
- Support for MacOS X

# Bug fixes (1 of 2)

---

- `sc_start()` after simulation has reached its internal maximum time value would overflow simulation time.
- `sc_trace` for `uint64`, `int64` missing
- `sc_set_time_resolution` not properly affecting VCD dump information.
- The value of `sc_clock` needs to be updated during update phase not evaluate (execution) phase to prevent race conditions.
- `sc_string` subscript operator may modify multiple instance because of copy semantics.
- Cpu risc example not shipped anymore

## Bug fixes (2 of 2)

---

- Error in `sc_bv` char constructor
- `sc_biguint` partial selection bug
- Missing terminating null char in `>>` operator for `sc_string`.
- The constructor `sc_module(const sc_module&)` is not defined
- Signal initialized in module CTOR not registered with its module.
- Deletion of main fiber should not occur in `~sc_cor_fiber`
- Need ability to compile with `Wno-deprecated`
- tracing ports after end of elaboration had no effect
- wait statements in `sc_module` ctor led to crashes