SystemC and SDL Co-Modelling Implementation

Nokia Research Center and Nokia Devices

Sergey Balandin, Ph.D.
Michel Gillet, Ph.D.

St. Petersburg State University of Aerospace Instrumentation

Alexey Rabin, Ph.D.
Valentin Olenev, Ph.D. student
Alexander Stepanov, M.Sc. student
Irina Lavrovskaya, M.Sc. student

Review of embedded system modelling

- Simulation features:
  - real-time process of system functionality check
  - sequence of basic events is kept
  - gets information about process state at certain moment of modelling time
- Purposes of modelling:
  - specification of embedded systems
  - verification of embedded systems
- Most using modelling languages:
  - SDL
  - SystemC
  - VHDL
  - Verilog, etc.

SystemC

- SystemC is a C++ library for modelling of embedded systems
- Provides additional interconnecting facilities: channels, events, ports
- Provides structuring facilities: modules, processes
- Provides possibility for synchronization

SDL

- SDL - Specification and Description Language
- SDL language is designed for the description of the structure and functionality of real-time systems
- The description of functionality of the system using SDL looks like a dependence of output signals from the input signals
- Structuring facilities (blocks and processes) give an ability to describe systems of all kind of difficulty

Purposes of SDL and SystemC

- Purposes of SDL:
  - specification and verification of embedded network protocols
  - specification of system architecture and system modules interconnections
  - development of models without dynamic setting of parameters

- Purposes of SystemC:
  - verification of embedded network protocols
  - specification of device functionality
  - development of models with dynamic setting of parameters

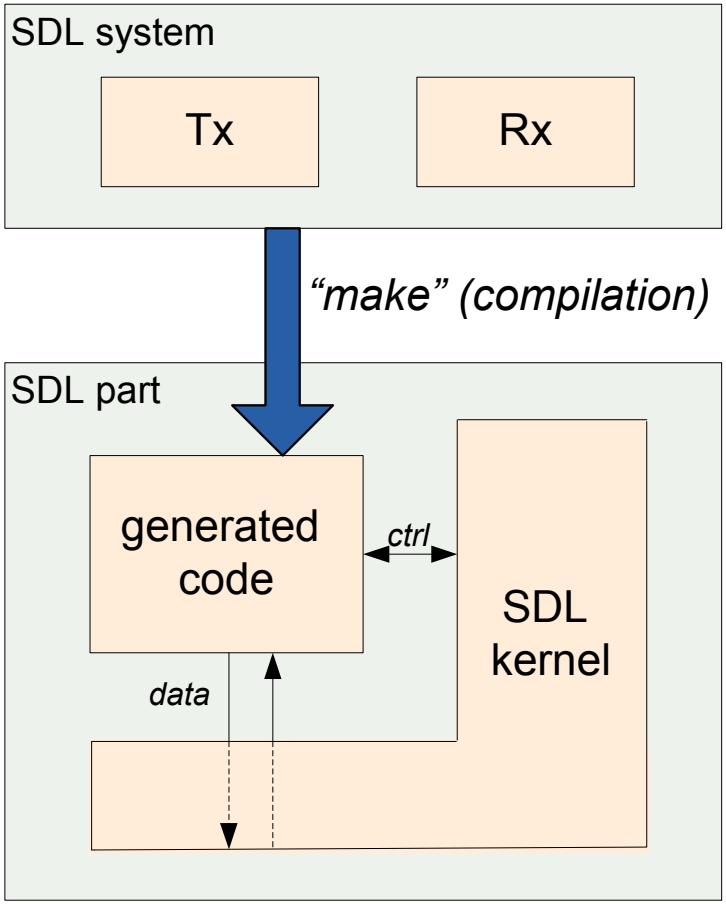
Organization of SDL and SystemC co-modelling

- Basic structure of SystemC/SDL co-model:
  - SystemC modelling core
  - SystemC part of model
  - SDL part of model (generated C/C++ code)

- SystemC modelling core provides:
  - scheduler
  - launcher
  - modelling time

Organization of SDL and SystemC co-modelling

- Requirements for SDL Tool:
  - possibility to generate C/C++ code from implemented SDL system
  - existing of SDL modelling core (in C/C++)
  - Main functions of SDL core:
    - initialization of system (*SDL_Init()*)
    - simulation (*SDL_Simulate()*)



Organization of SDL and SystemC co-modelling

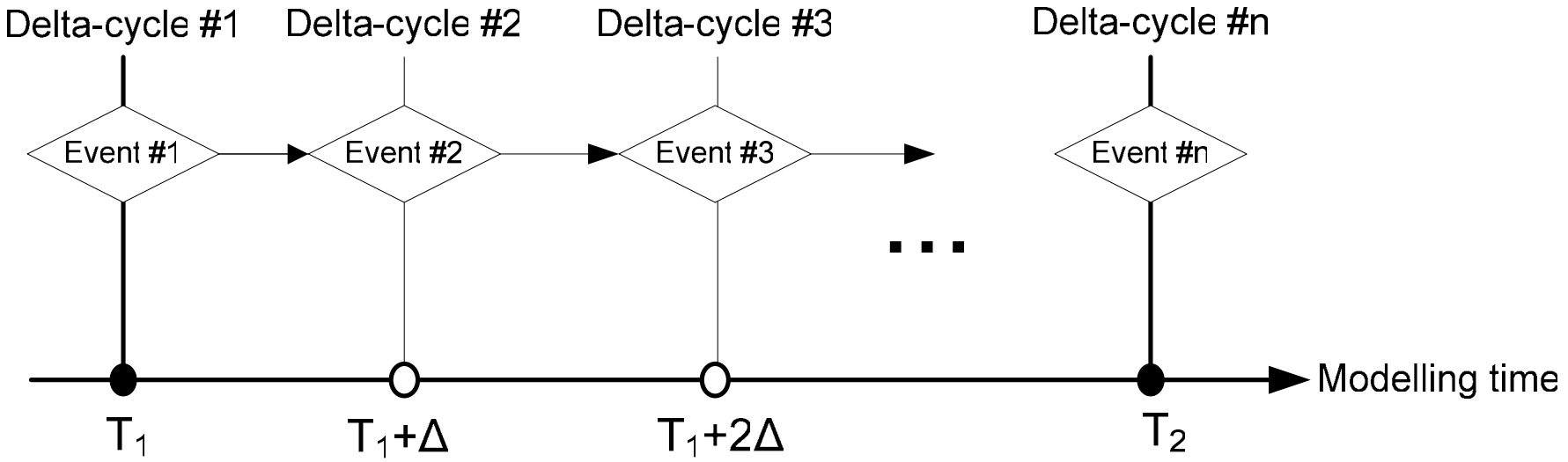
- SDL/SystemC co-modelling implementation workflow:
  - preparation of SDL system as the part of the whole model
  - generation of C/C++ code on the basis of created SDL system
  - insertion of generated C/C++ code into the SDL core
  - preparation of SystemC part of the model
  - integration of SDL core with the generated C code into the whole model

Organization of SDL and SystemC co-modelling

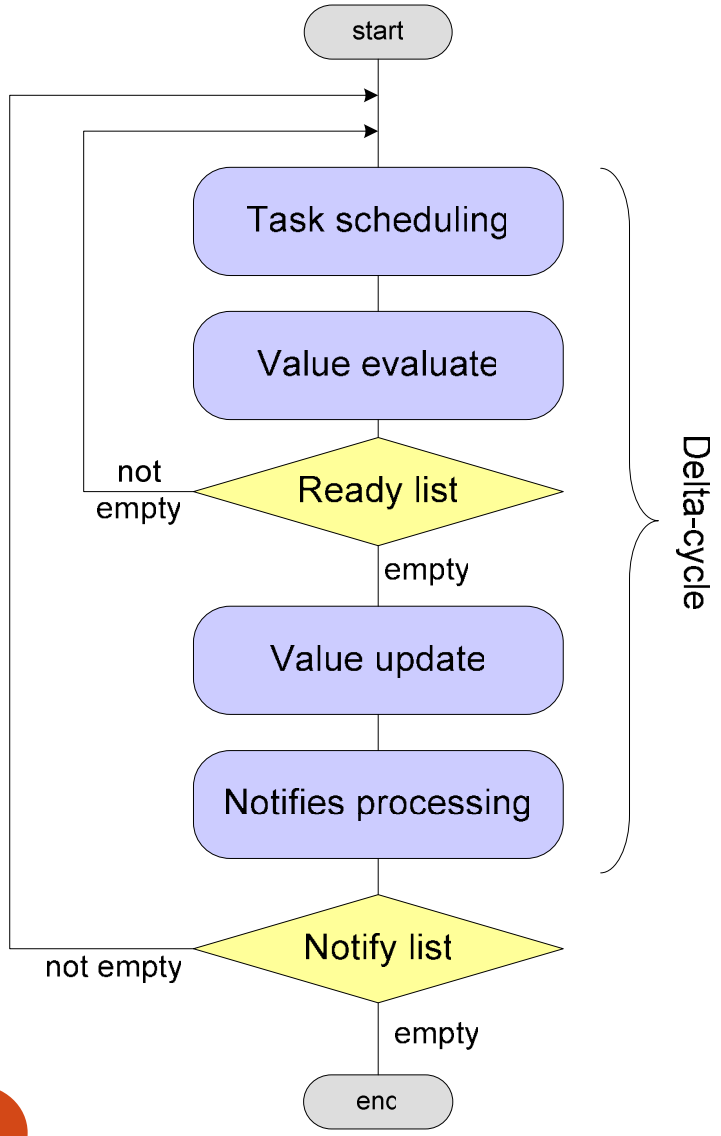
- Advantages of approach:
  - work with both SDL and SystemC parts at the same time is possible
  - usage of SystemC modelling core for SDL simulation is possible
- Disadvantages of approach:
  - special interface between C/C++ analogue of SDL and SystemC models is necessary
  - structural changes could be implemented only through SDL code using

SystemC scheduling

- Common scheduler for both SDL and SystemC parts is framework of co-modelling organization
- Main notions of SystemC scheduling:
  - delta-cycle
  - delta-delay



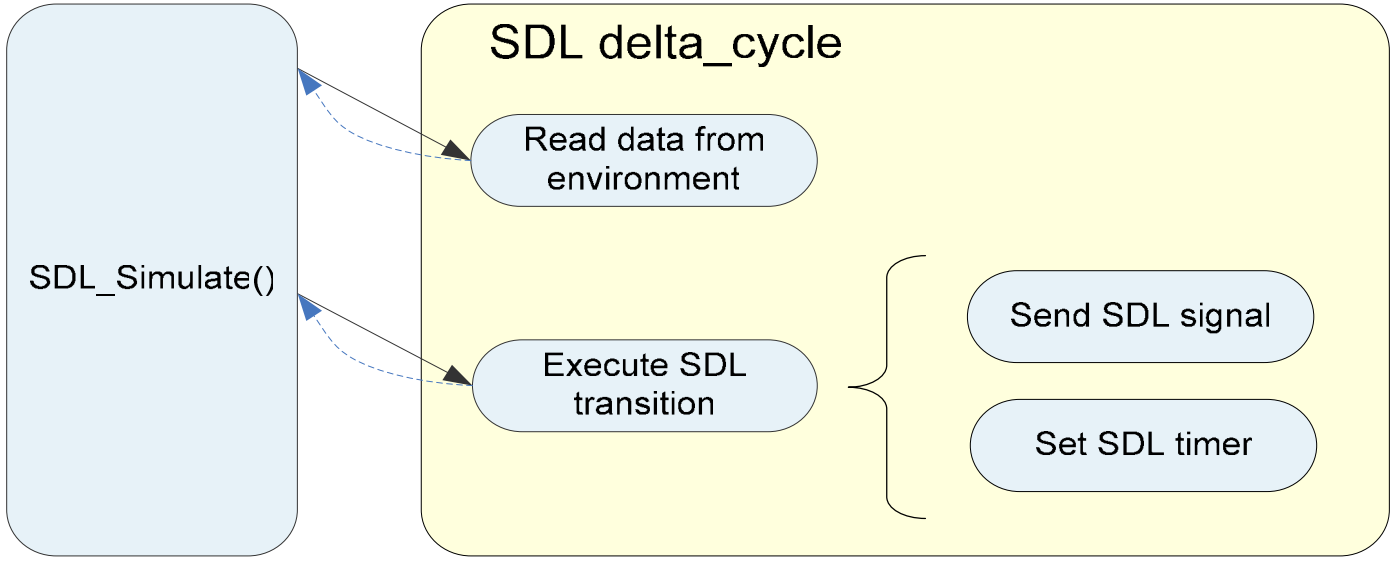
SystemC scheduling



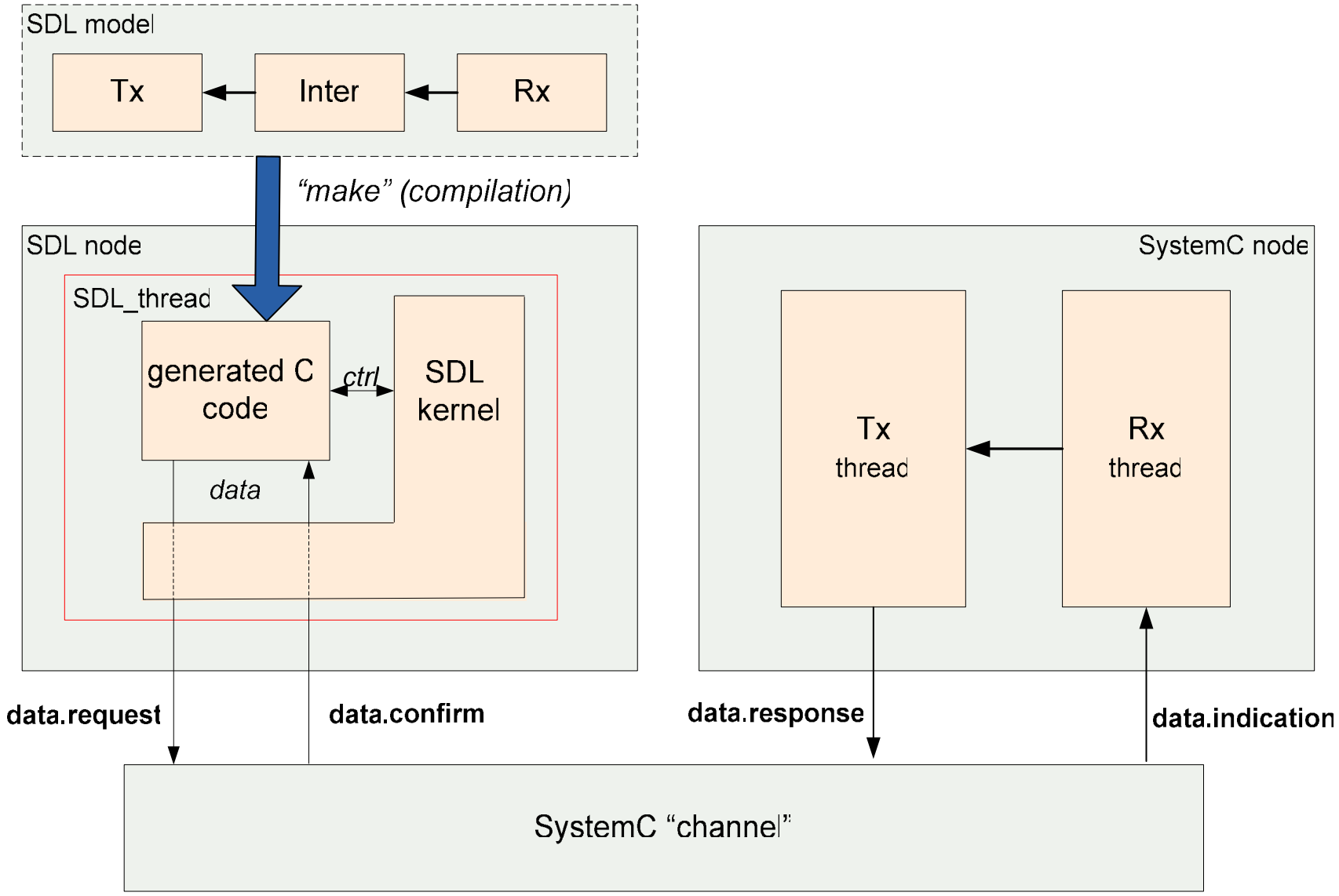
- Workflow of SystemC scheduler:
  - ***start*** – start of next clock-cycle
    - ***task scheduling*** – launch of next ready process execution
    - ***value evaluate*** – calculation of new values
    - ***value update*** – assignment of calculated values
    - ***notifies processing*** – search of events, which should be processed at next delta-cycle
  - ***end*** – end of current clock-cycle

SDL scheduling

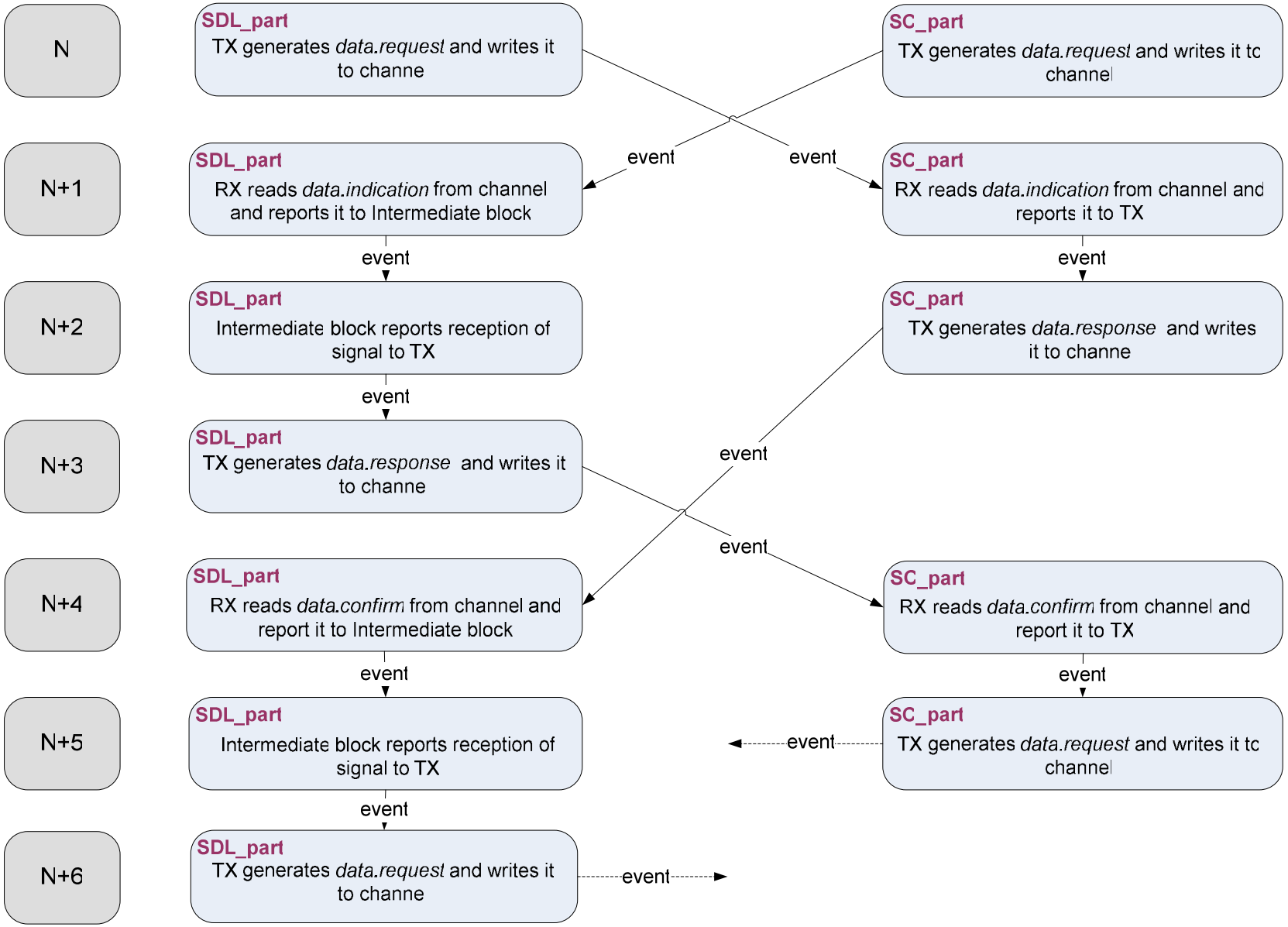
- Each SDL delta-cycle contains only one execution of SDL transition
- Ways for SDL events scheduling:
  - signal – new event should be performed at the current moment of modelling time
  - timer – new event should be performed at certain moment of modelling time, after all current time events will be performed



Sample of SDL/SystemC co-modelling implementation



Sample of SDL/SystemC co-modelling implementation



Conclusions

- SDL and SystemC co-modelling is based on:
  - compilation of SDL part of model into C/C++ code
  - usage of SystemC scheduler as a common scheduler for both parts of model
  - usage of SystemC modelling time as a common modelling time for both parts of model
  - organization of single modelling system for both SDL and SystemC parts

Thank you