TECHNISCHE UNIVERSITÄT WIEN
Vienna | Austria

Institut für Computertechnik
Institute of Computer Technology

VELS: VHDL E-Learning System

Martin Mosbeck, Daniel Hauer, Axel Jantsch
Institute of Computer Technology,
TU Wien, Vienna, Austria

NorCAS 2018, Tallinn, 2018-10-31

What is VELS?

- VHDL E-Learning System
- Automatic task generation, assessment, feedback
- Communication via email
- Open source @ GitHub

Why VELS?

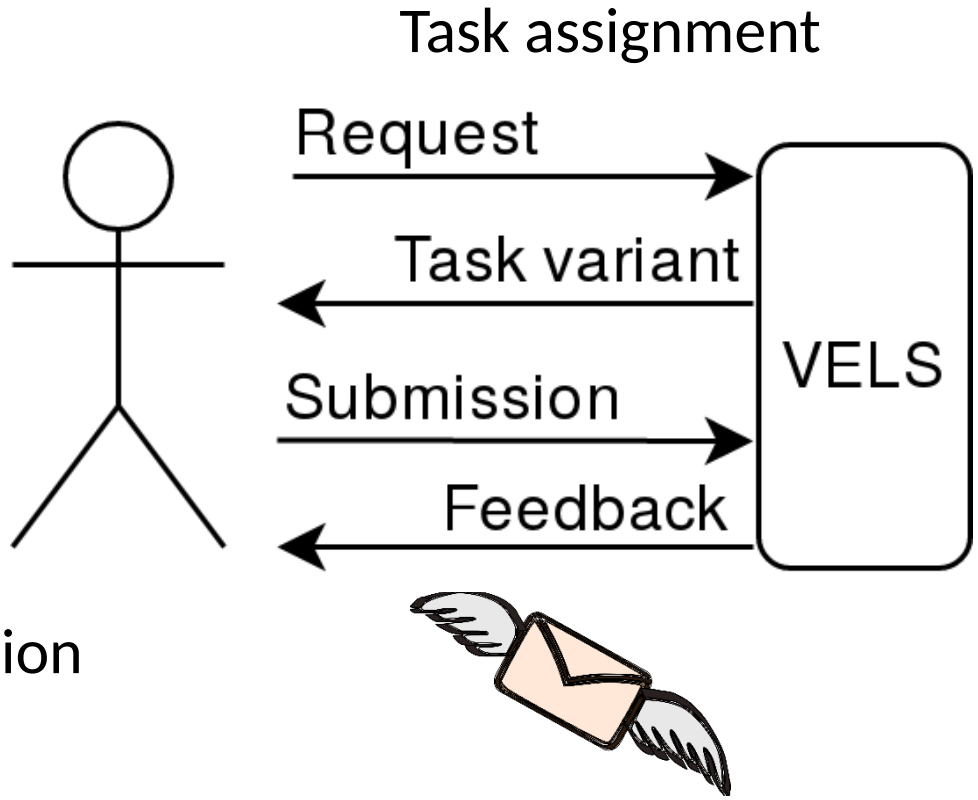
- Large classes (250 students/semester)

- Benefits for students:
  - practice with adequate tasks
  - frequent, instant feedback
  - freedom of space, pace, study time scheduling

- Benefits for teachers:
  - framework to develop, deploy tasks
  - automatic assessment
  - progress monitoring

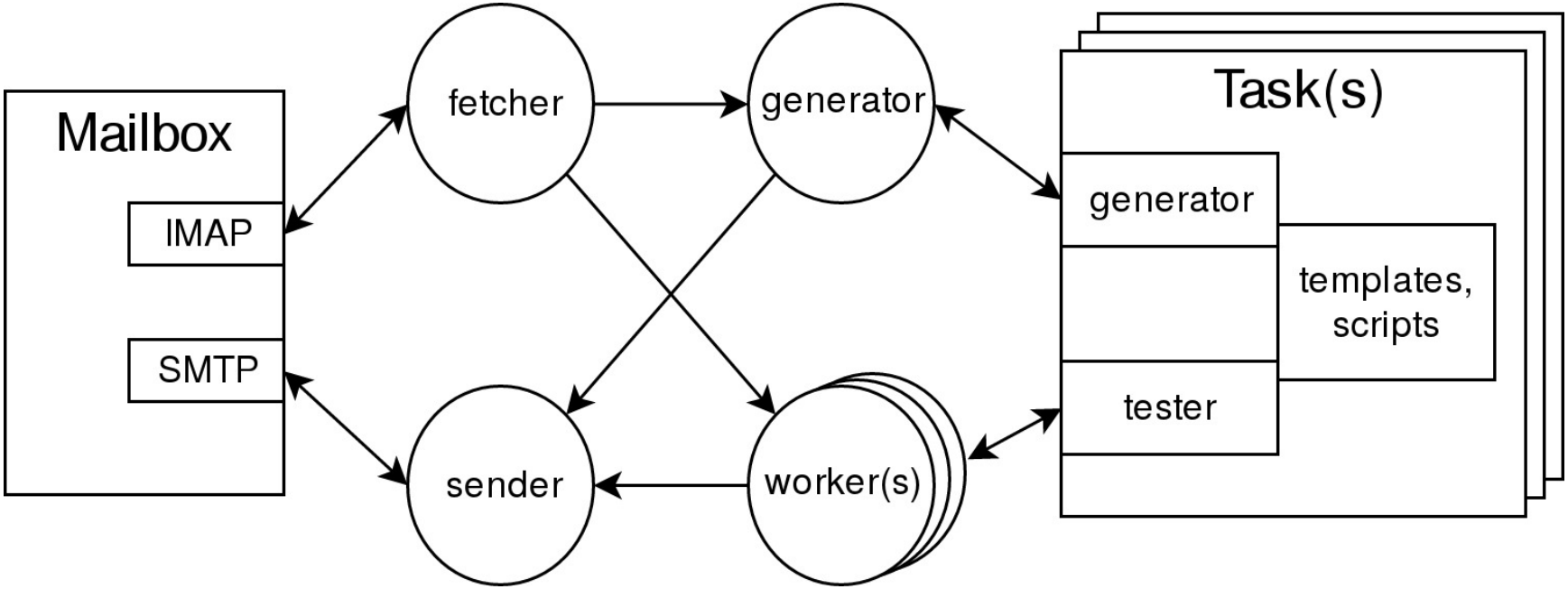
Key features

- Uniform task system
- Parameterized tasks



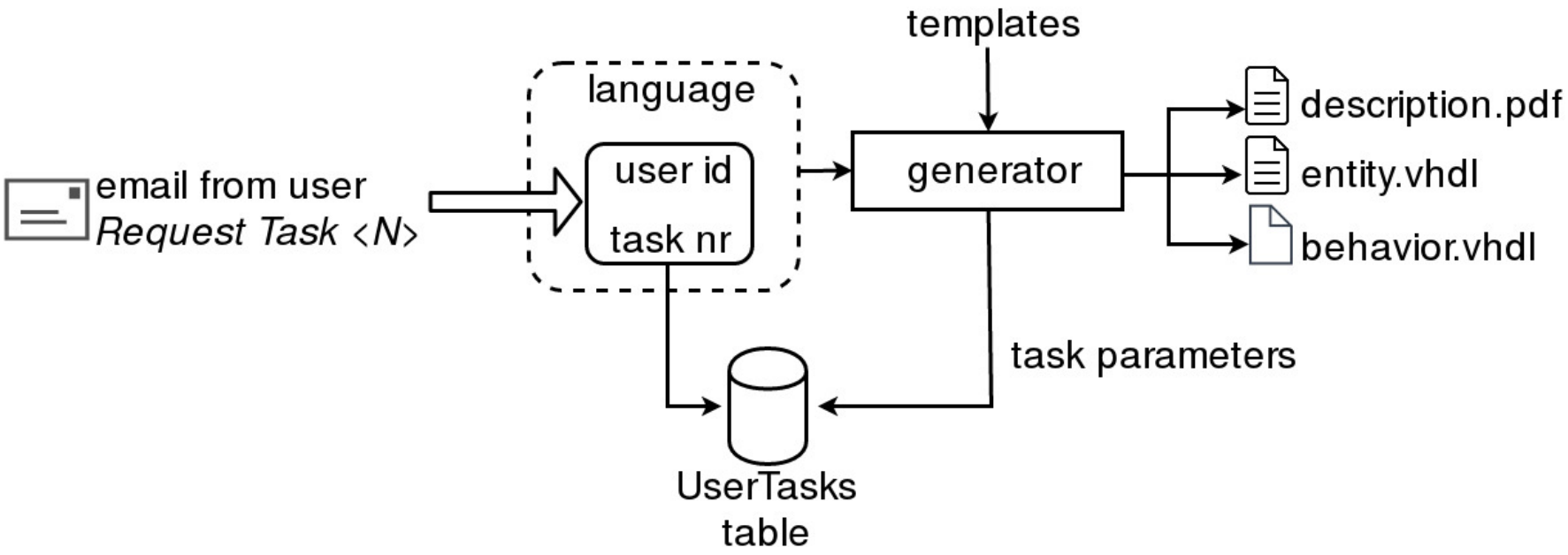
- New test for each submission
- Simulator interface
- Communication student < -- > VELS via email

Autosub submission system & task system



- Multi-threaded python deamon
- Periodically fetches new emails

Task request



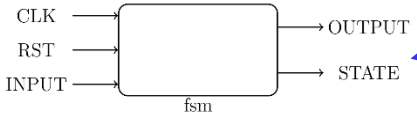
- Request generates random variant of task
- Sent to user as email

Example task - Finite State Machine

Mealy FSM (Finite State Machine)

Your task is to program the behavior of an entity called "fsm". This entity is declared in the attached file "fsm.vhdl" and has the following properties:

- Input: CLK with type std_logic
- Input: RST with type std_logic
- Input: INPUT with type std_logic_vector of length 2
- Output: OUTPUT with type std_logic_vector of length 2
- Output: STATE with type fsm_state



Do not change the file "fsm.vhdl"!

The "fsm" entity shall have the behavior of a deterministic **MEALY** finite state machine with the following constraints:

- Behavior according to state diagram as seen in Figure 1. The inputs transit to the next state and the output in the next state is labelled on the edges.
- Transitions with rising edge of the clock (port CLK, this is a square-wave clock signal).
- Synchronous design: all new outputs (port STATE and port OUTPUT) have to be set with rising edges of the clock.
- The current state has to be output at the output port STATE.
- At synchronous reset (port RST=1, rising edge of clock port CLK) the state machine is set to an initial state: STATE= START, OUTPUT= 00

This behavior has to be programmed in the attached file "fsm_beh.vhdl".

The type fsm_state is declared in the attached "fsm_package.vhdl" file. The needed package to use this type is already imported in the attached "fsm_beh.vhdl" and "fsm.vhdl".

interface

block diagram

description text

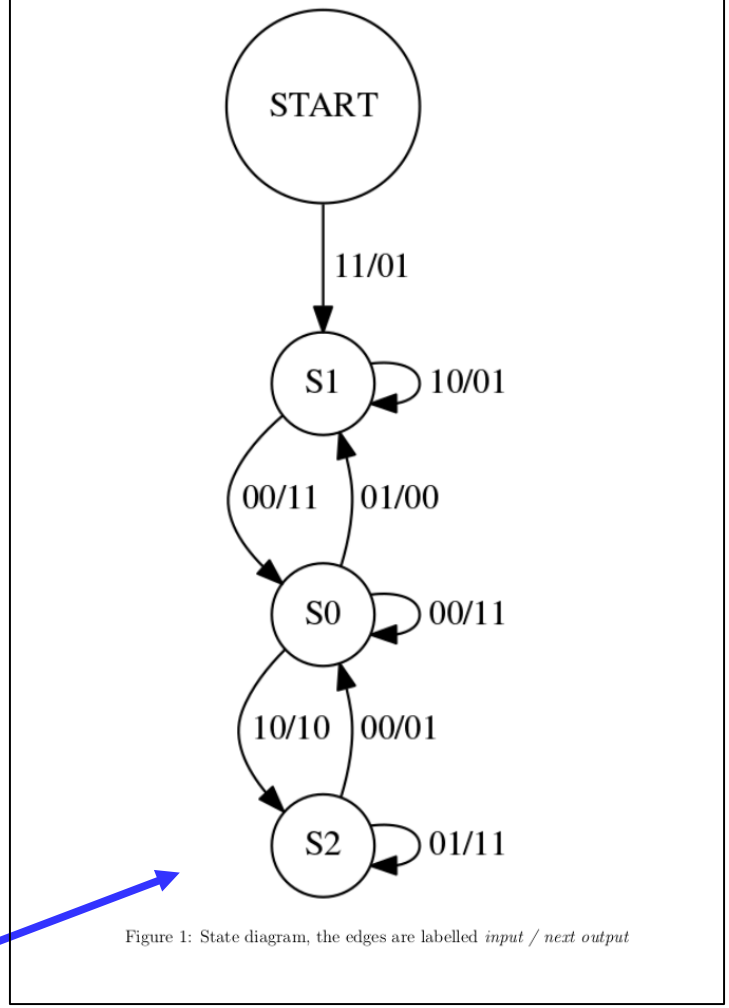
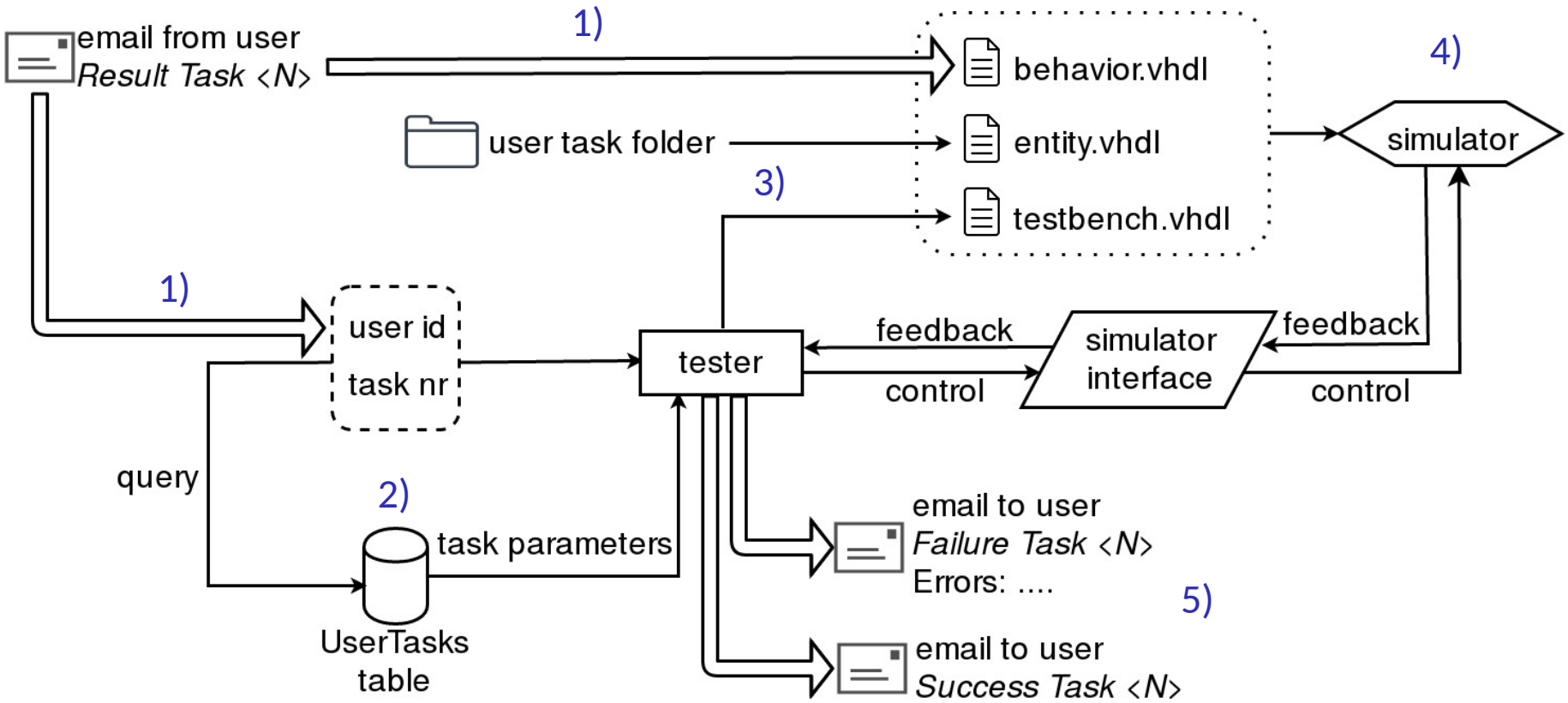


Figure 1: State diagram, the edges are labelled *input / next output*

supporting figure

Task submission



TaskNr	Task Start	Task Deadline	Task Name
1	2018-09-18 16:00	2018-10-10 16:19	gates
2	2018-09-18 16:00	2018-10-10 16:19	demux
3	2018-09-18 16:00	2018-10-15 16:19	counter
4	2018-09-18 16:00	2018-10-17 16:19	fsm_mealy
5	2018-09-18 16:00	2018-10-17 16:19	ROM
6	2018-09-18 16:00	2018-10-17 16:19	RAM
7	2018-09-18 16:00	2018-10-17 16:19	pwm
8	2018-09-18 16:00	2018-10-17 16:19	arithmetic

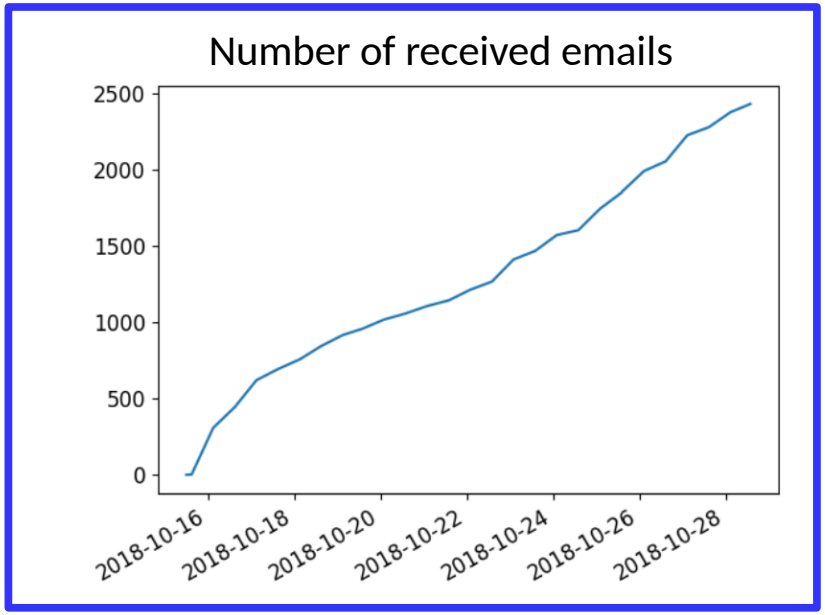
Task Statistics

Task Id	Number Submissions	Number Successful	Success Percentage
1	837	228	27%
2	636	188	29%

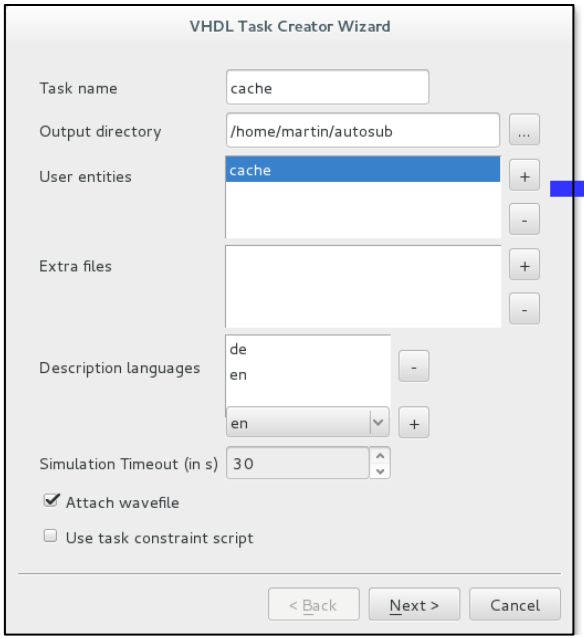
Statistics Counters

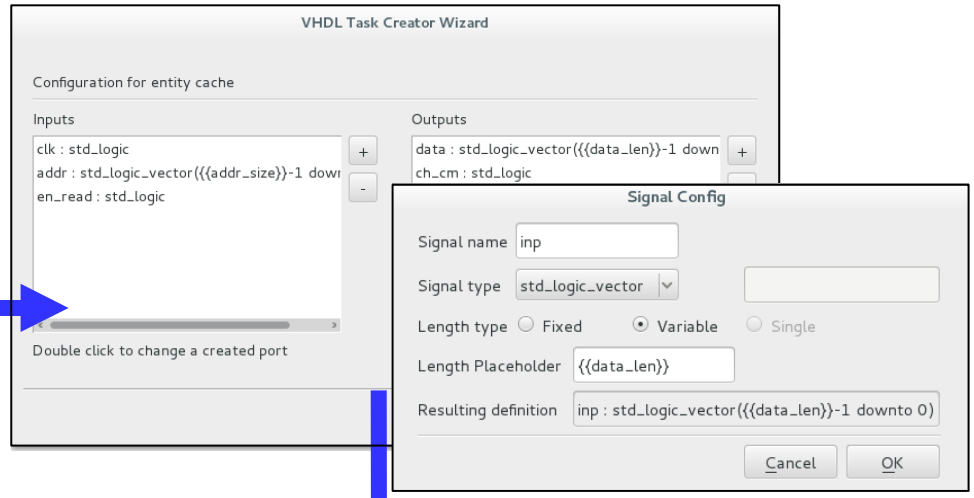
Name	Value
nr_mails_fetched	2694
nr_mails_sent	2878
nr_non_registered	74
nr_questions_received	16
nr_status_requests	45

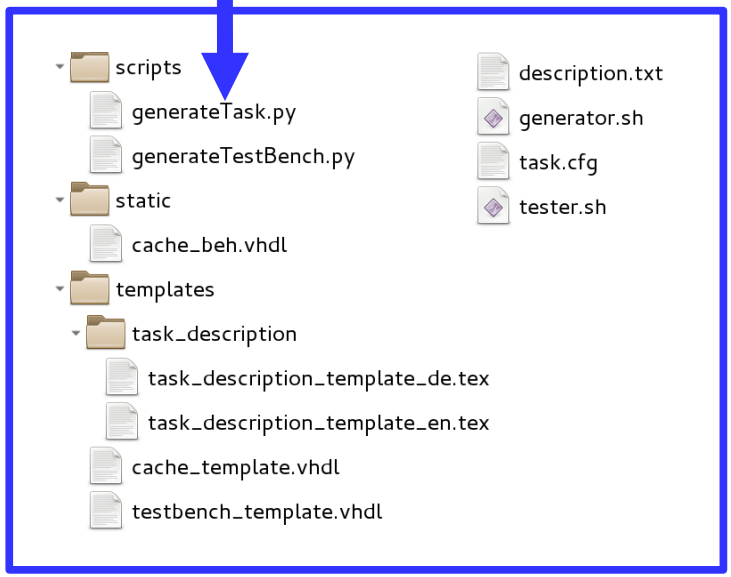
- Web interface
- Configure tasks, whitelist
- Monitor progress



Task generator tool







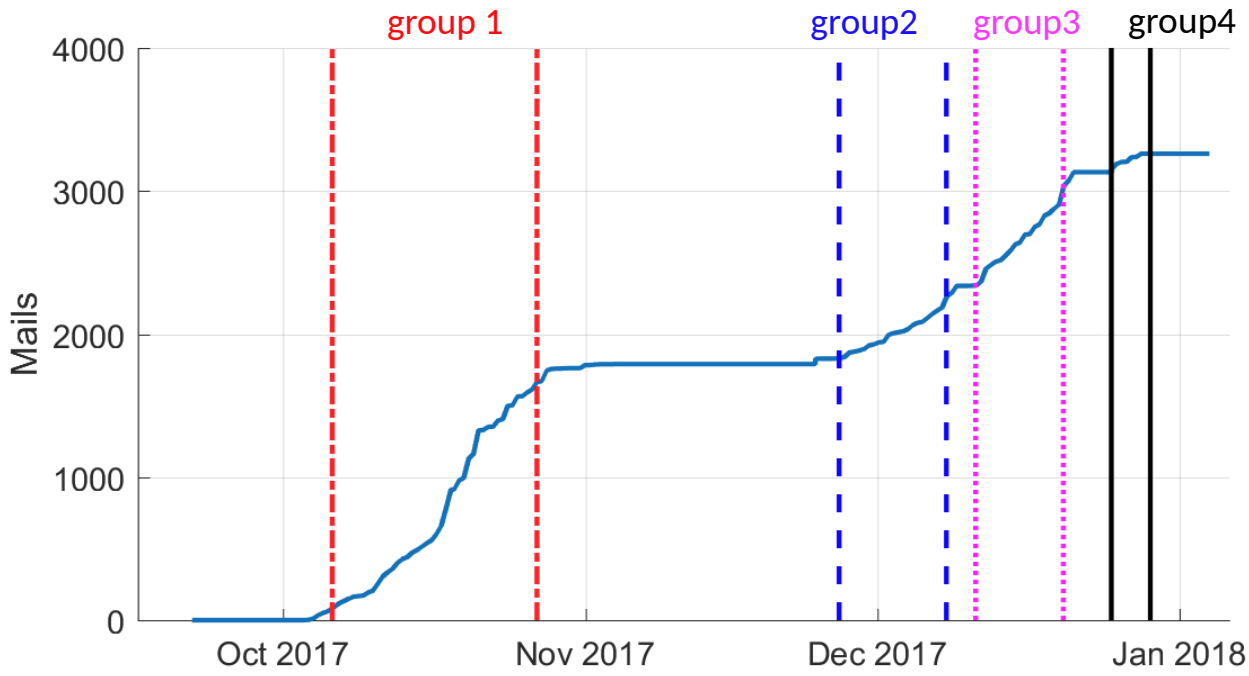
skeleton task

Allows to focus on:

- task description
- task variant generation
- testbench generation

Deployment & Results (1/2)

- 3 years, 2 courses/each winter semester
- Overall 1000 students



Mandatory, continuous learning in the graduate course „Digital Integrated Systems“ (50 students)

Deployment & Results (2/2)

Task nr.	Students			Submissions		
	Sum	Passed	Rate	Sum	Passed	Rate
1	242	195	81 %	1215	317	26 %
2	242	185	76 %	905	326	36 %
3	242	188	78 %	623	294	47 %
4	242	176	73 %	1967	313	15 %
5	242	160	66 %	1318	240	18 %
6	242	84	35 %	1411	115	8 %
7	242	147	61 %	795	247	31 %
8	242	80	33 %	1805	141	7 %

Free practice in undergraduate course „Microcomputer“

Conclusions & take away

- 3 years use, open source @ GitHub
  https://github.com/autosub-team/autosub

- Allows continuous, incremental learning
- Human teaching still needed
- Clear task specification & comprehensive feedback!

- Synthesis tool interface
- Moodle interface, bundle as Debian package

Questions?