Using SIL Arithmetic to Design Safe and Secure Cyber-physical Systems

Raimund Kirner
University of Hertfordshire

MS@CPS Open Info Day, 26.08.2020

Cyber-physical Systems

- CPS: Networked embedded systems
- Systems of systems
- Services of mixed criticality

Increasing Flexibility for Building Cyber-physical Systems

- Building system services from components that are **less rigorously** developed than **required** by the domain-specific safety standard.

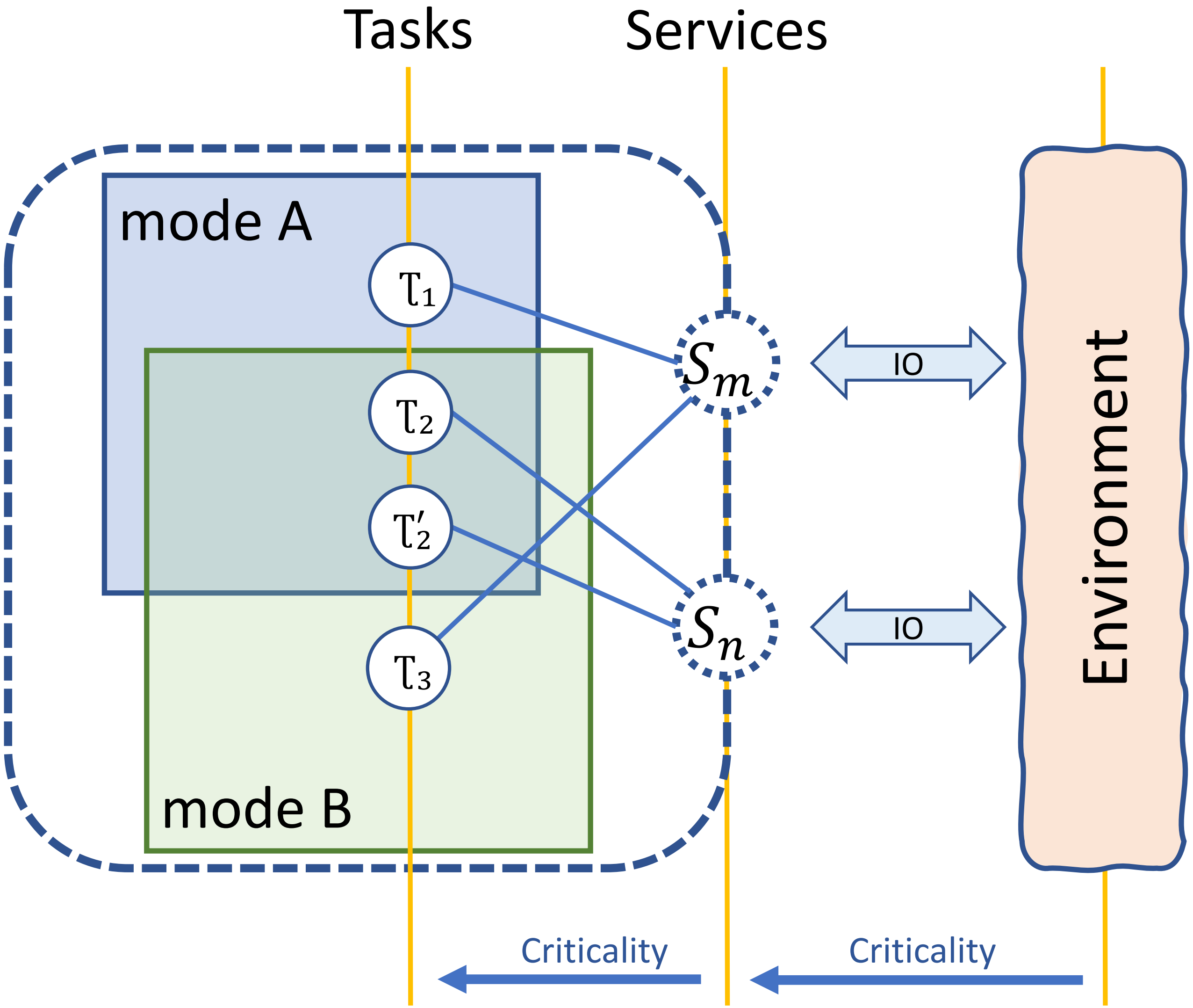
- Why would we want to do that?
  - simplification of development
  - cost efficiency

What is Mixed Criticality?

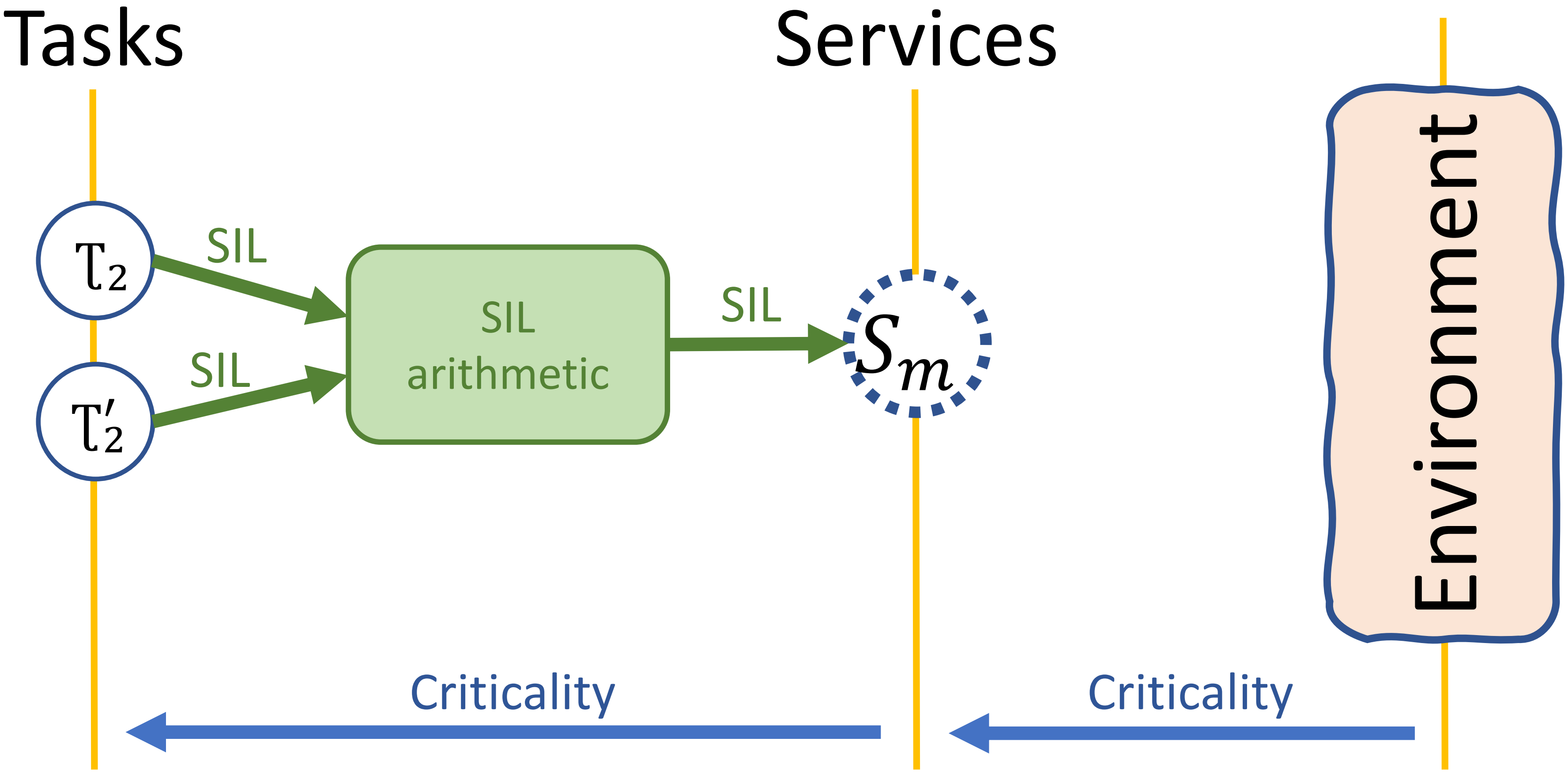
- CRIT (S_1) > CRIT (S_2) ➔
  - "service S_1 is more critical than S_2 for the mission"
  - safety integrity level (SIL) according to domain-specific safety standards (IEC 61508, DO-178b, ISO 26262, etc.):

    SIL(S_1) >= SIL(S_2)
  - assurance level of S_1 is higher than of S_2 Example (*Vestal,RTSS'07*, criticality LO/HI): higher timing assurance available for service S_1 than for S_2

System Model: Services & Tasks



The Principle of SIL Arithmetic



Example: Unmanned Aerial Vehicle (UAV)

- System with 4 services
- Service S_3 realised with 2 tasks, each SIL 1 (using SIL Arithmetic)

Service (Task)	*Description*	*SIL*
S_1 (τ_1)	trajectory	3
S_2 (τ_2)	earth monitoring	2
S_3 (τ_3 and τ_3')	communication with station	2
S_4 (τ_4)	logging of tasks' events	1

Example: Unmanned Aerial Vehicle (UAV)

- Tasks before failure:

Service (Task)	*Description*	*SIL*
S_1 (τ_1)	trajectory	3
S_2 (τ_2)	earth monitoring	2
S_3 $(\tau_3$ and $\tau_3')$	communication with station	2
S_4 (τ_4)	logging of tasks' events	1

Example: Unmanned Aerial Vehicle (UAV)

- Tasks after failure of task τ_3:
  Service S_3 only provided by task τ_3'

Service (Task)	*Description*	*SIL*
S_1 (τ_1)	trajectory	3
S_2 (τ_2)	earth monitoring	2
S_3 $(\tau_3'$ only)	communication with station	1
S_4 (τ_4)	logging of tasks' events	1

Example: Unmanned Aerial Vehicle (UAV)

While assurance level of S3 after the failure of t3 is reduced from **SIL2** to **SIL1**, the mixed criticality scheduler must treat the service S3 based on its original application-dependent criticality

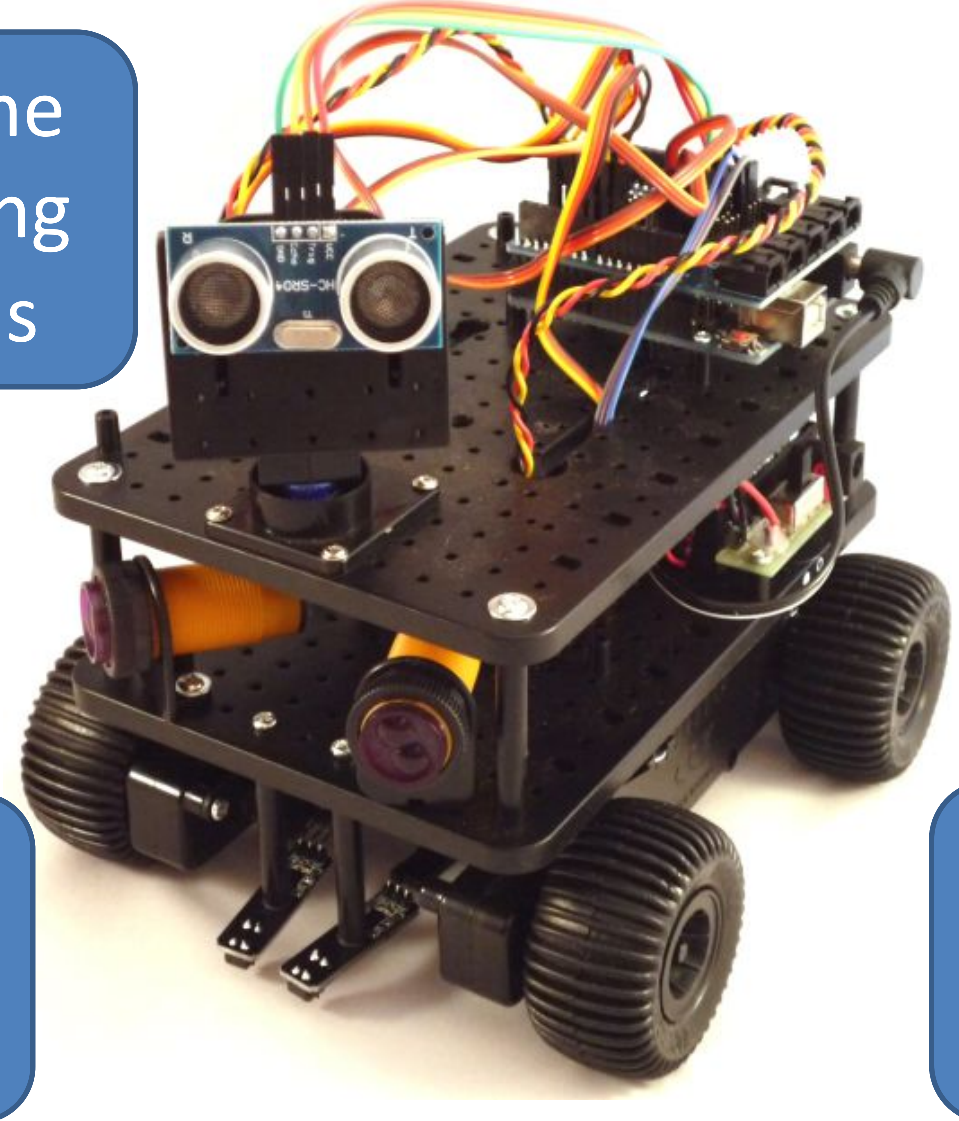
→ **scheduler** should treat task t3’ with **increased importance** to achieve this

S_1 (τ_1)	trajectory	3
S_2 (τ_2)	earth monitoring	2
S_3 $(\tau_3'$ only$)$	communication with station	1
S_4 (τ_4)	logging of tasks’ events	1

Conclusion

- Discussion of SIL arithmetic: its motivation and usage
- Argumentation why mixed-criticality schedulers should be aware of underlying use of SIL arithmetic: to maintain assurance level of service
- Work to be done: development of SIL arithmetic aware mixed-criticality schedulers

Case-study Driven Education of Cyber-physical Systems



Real-time Operating Systems

Resilient Computing

Use of Sensors

System Programming

Feedback-based Control