Real-Time (Paradigms) (35a)

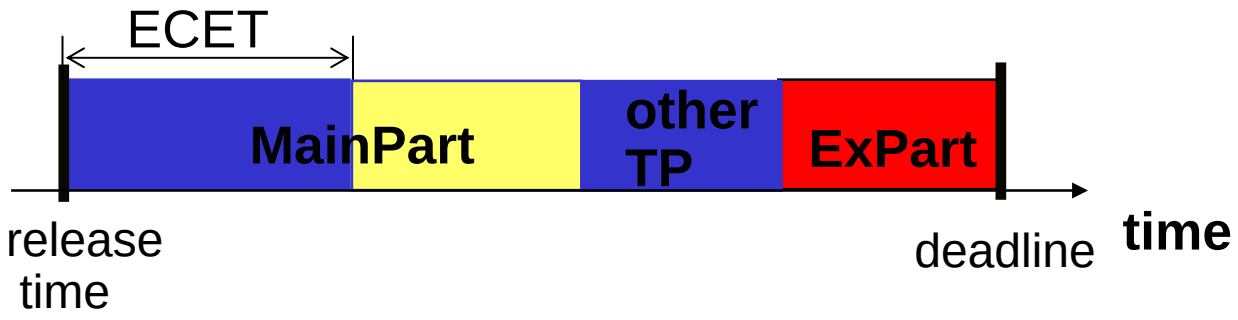
Goals of TAFT (Time - Aware Fault - Tolerant) Scheduling

- No Handling of tasks with unknown or too pessimistic WCETs
  - Introduction of Expected Case Execution Time (ECET)
- Still with Timing Guarantees
  - Scheduled exception handling **before** the deadline
- Fault-Tolerance with respect to timing errors
  - Graceful degradation in overload situations
  - Tradeoff between functionality and timing

Real-Time (Paradigms) (35b)

TAFT Scheduling

- Each module is scheduled as a task pair consisting of a main part and an exception part
  - Main part: actual module functionality, ECET scheduled
  - Exception part: module specific exception handling, WCET scheduled
- Timing faults are confined to modules



Real-Time (Paradigms) (36)

Three-level Scheduling:

•Level One – ExceptionParts

–Highest dispatching priorities

–LRT (Latest Release Time - Reverse-EDF)

–Tries to do everything as late as possible

•Level Two - MainParts executed within the reserved (and guaranteed) ECET

–Medium dispatching priorities

–EDF

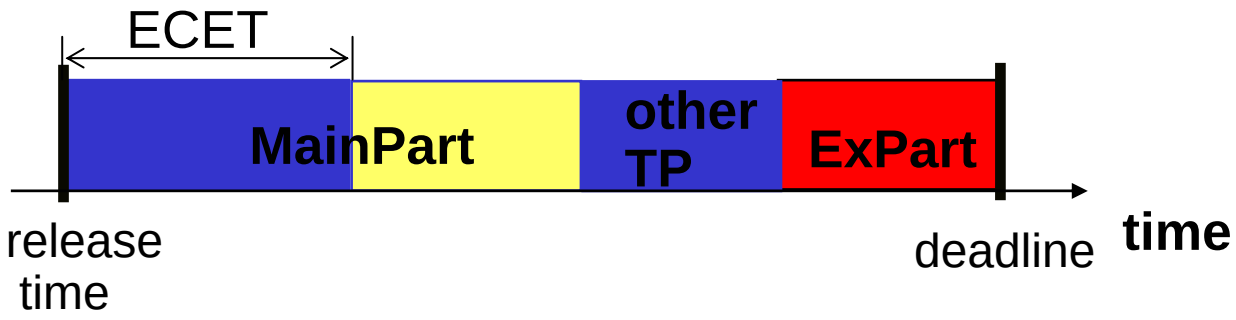
–Tries to do everything as soon as possible

•Level Three - MainParts executed beyond the reserved (and guaranteed) ECET

–Lowest dispatching priorities

–EDF

–Tries to do everything as soon as possible



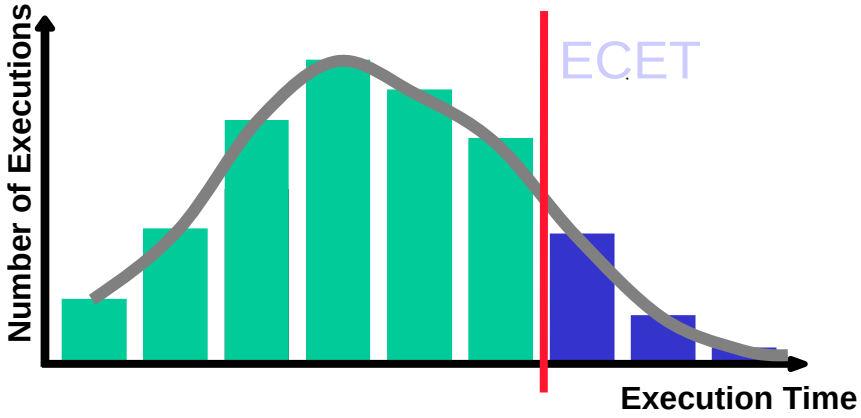
Real-Time (Paradigms) (37)

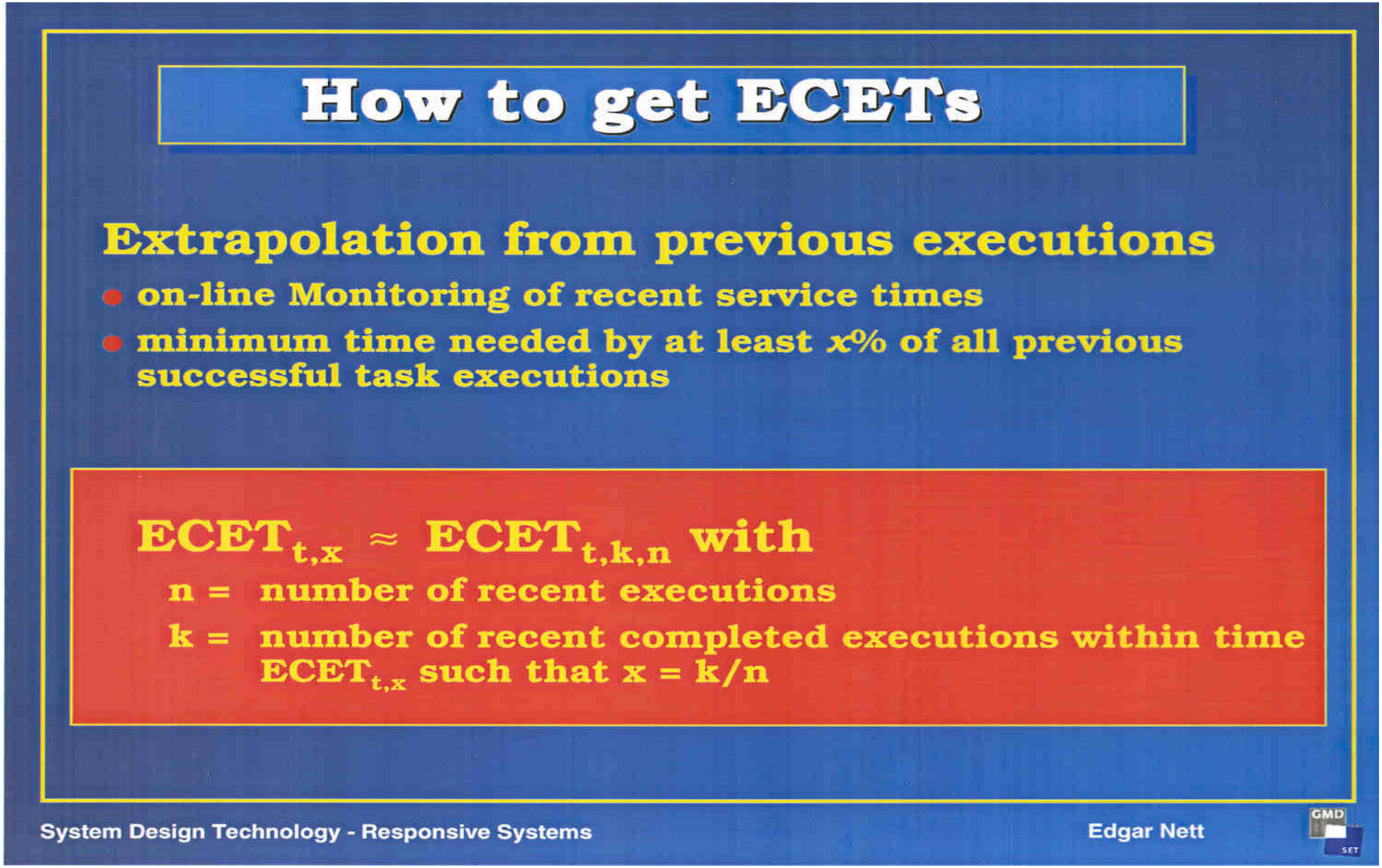
$ECET_{t,p}$ of task-instance t of task τ with probability p

- CPU-time required to complete task-instance t with probability p
- p-quantile of the probabilistic density function of τ's execution time

$ECET_{t,k,n}$ - The minimal execution time that was needed to successfully complete at least k out of the last n most recent executions of τ before t.

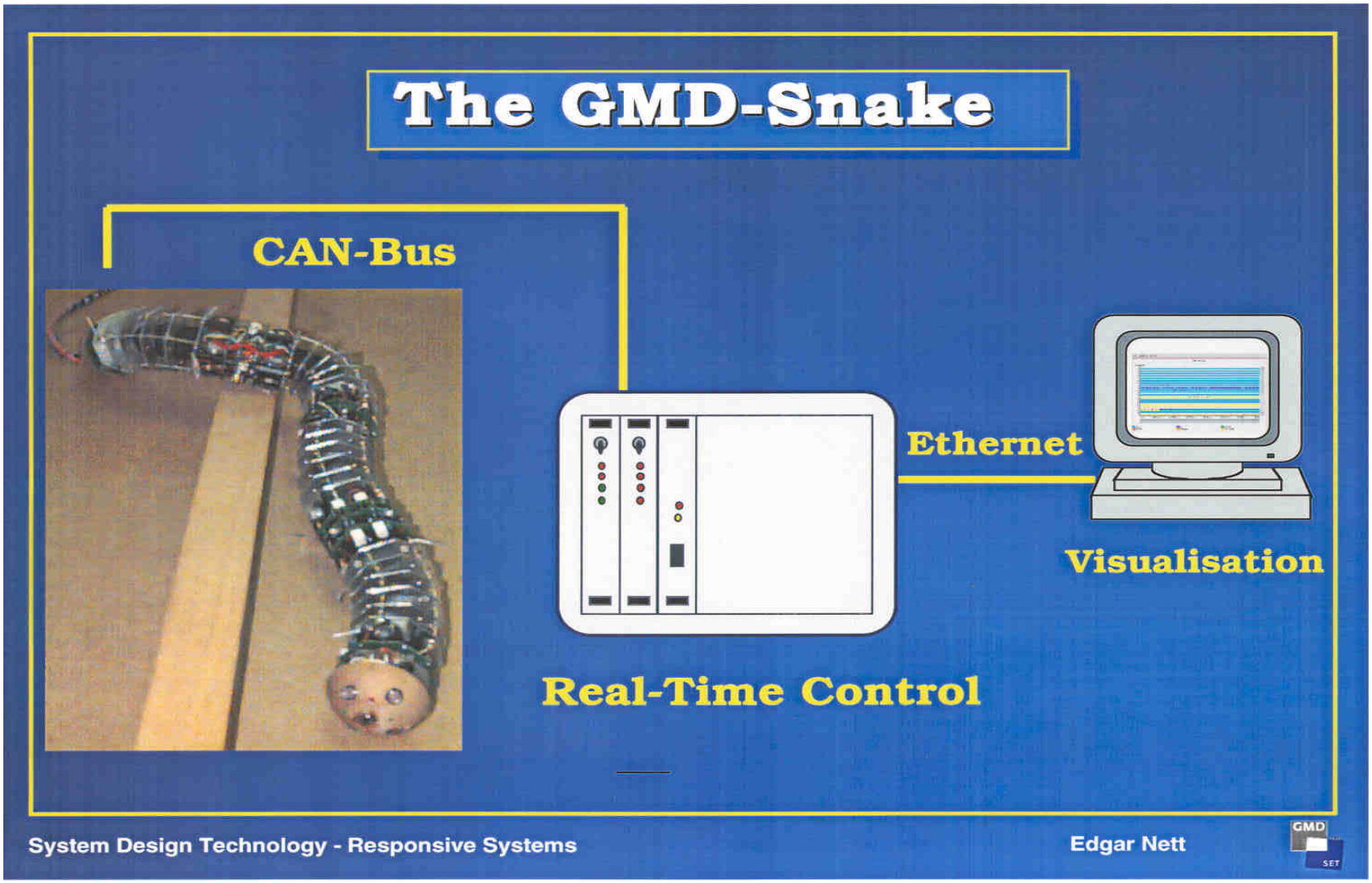
- A statistic quantity



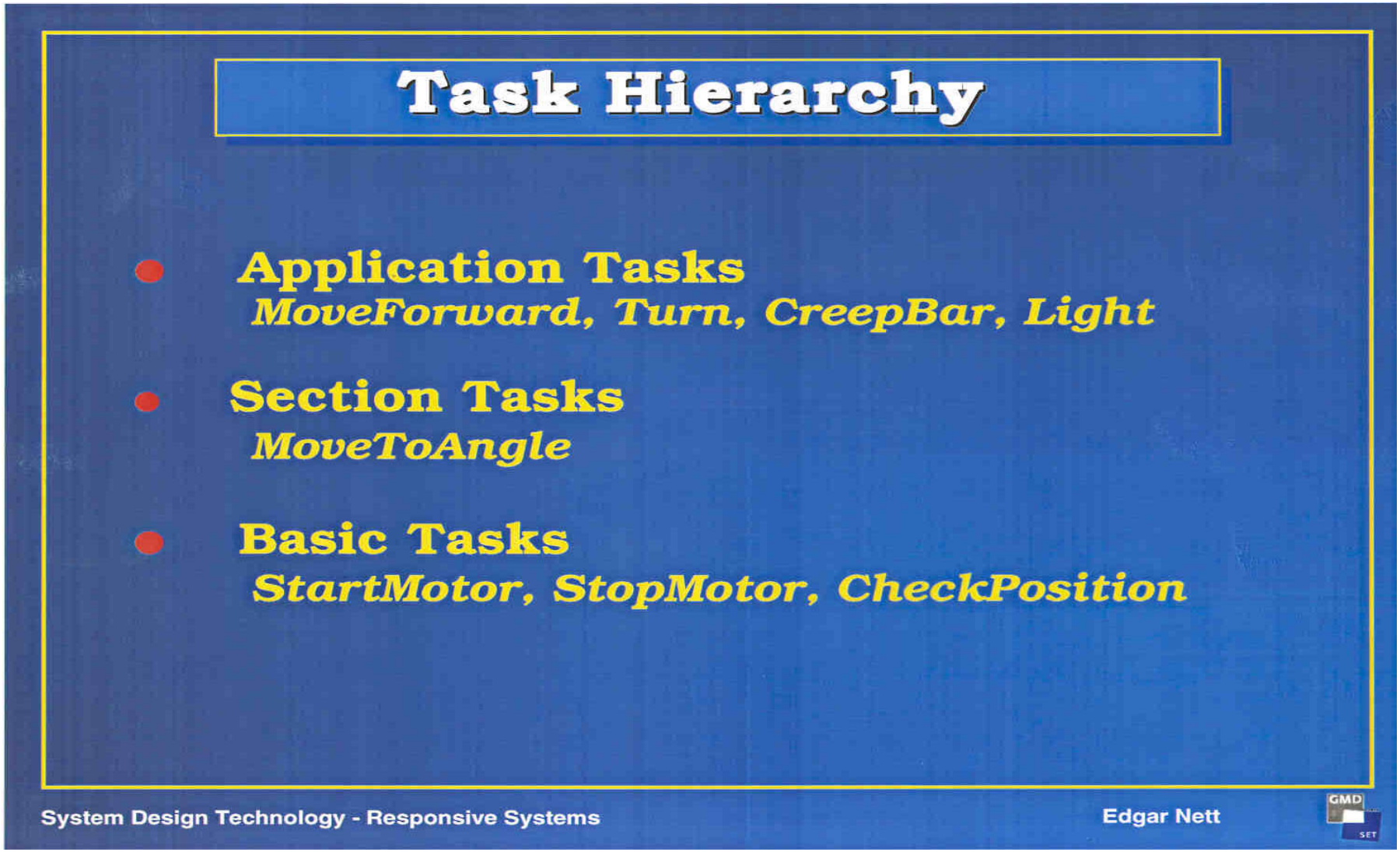

How to get ECETs
Extrapolation from previous executions
on-line Monitoring of recent service times
minimum time needed by at least x% of all previous successful task executions
$ECET_{t,x} \approx ECET_{t,k,n}$ with
n = number of recent executions
k = number of recent completed executions within time $ECET_{t,x}$ such that x = k/n
System Design Technology - Responsive Systems
Edgar Nett
GMD
SET

External Reviewers

Real-Time (Paradigms) (39a)



Real-Time (Paradigms) (40)



Real-Time (Paradigms) (41)

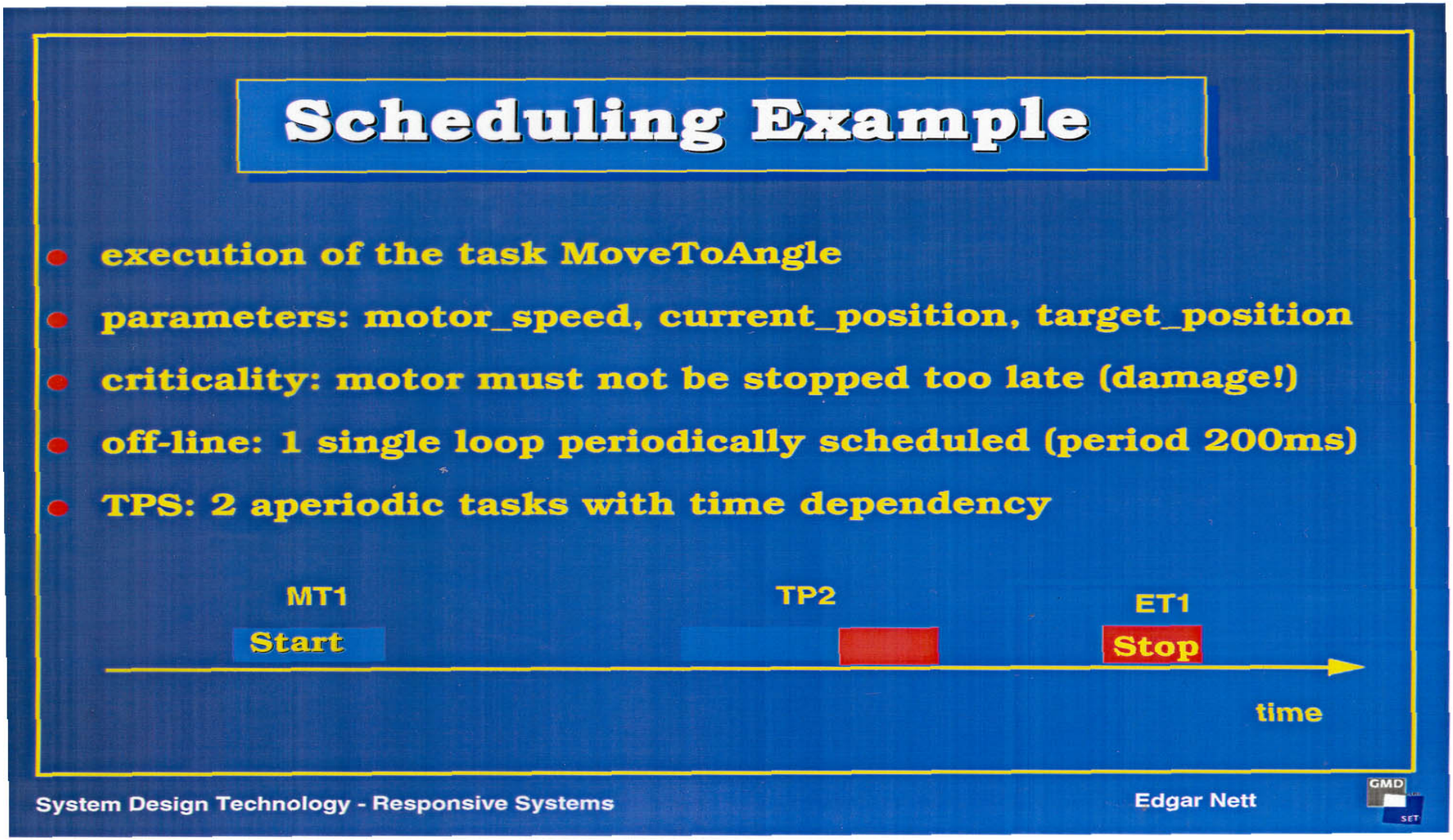
Timing Constraints:

- Calibration messages each 40 ms
- Polling angle sensor each 80 ms
- StopMotor (situation dependent)

Task Classification (Example):

- Hard: Calibration task
- Critical: MoveToAngle task
- Soft: Light task

Real-Time (Paradigms) (42)



Real-Time (Paradigms) (43)

Example (Cont´d)

Task: MoveToAngle (45 degrees)

WCET for Start resp. CheckPositon: 16 ms

computed duration for move: 2 sec

Resource consumption:

off-line: 10 x 16 ms = 160 ms within 2 sec reserved

TPS: 2 x 16 ms = 32 ms within 2 sec reserved

System Design Technology - Responsive Systems Edgar Nett

Real-Time (Paradigms) (44)

Example (Cont´d)

Problem: motor speed is not constant

depends on: direction (up, down, left, right), current position, and simultaneous moves in other sections

- **off-line: worst case approach, i.e. maximum speed has to be assumed)**
- **TPS: on-line adaptation possible: if motor would stop too early,then, schedule an additional CheckPosition, if possible**

System Design Technology - Responsive Systems Edgar Nett

Real-Time (Paradigms) (45)