Design methods and tools for real-time (automotive) embedded systems

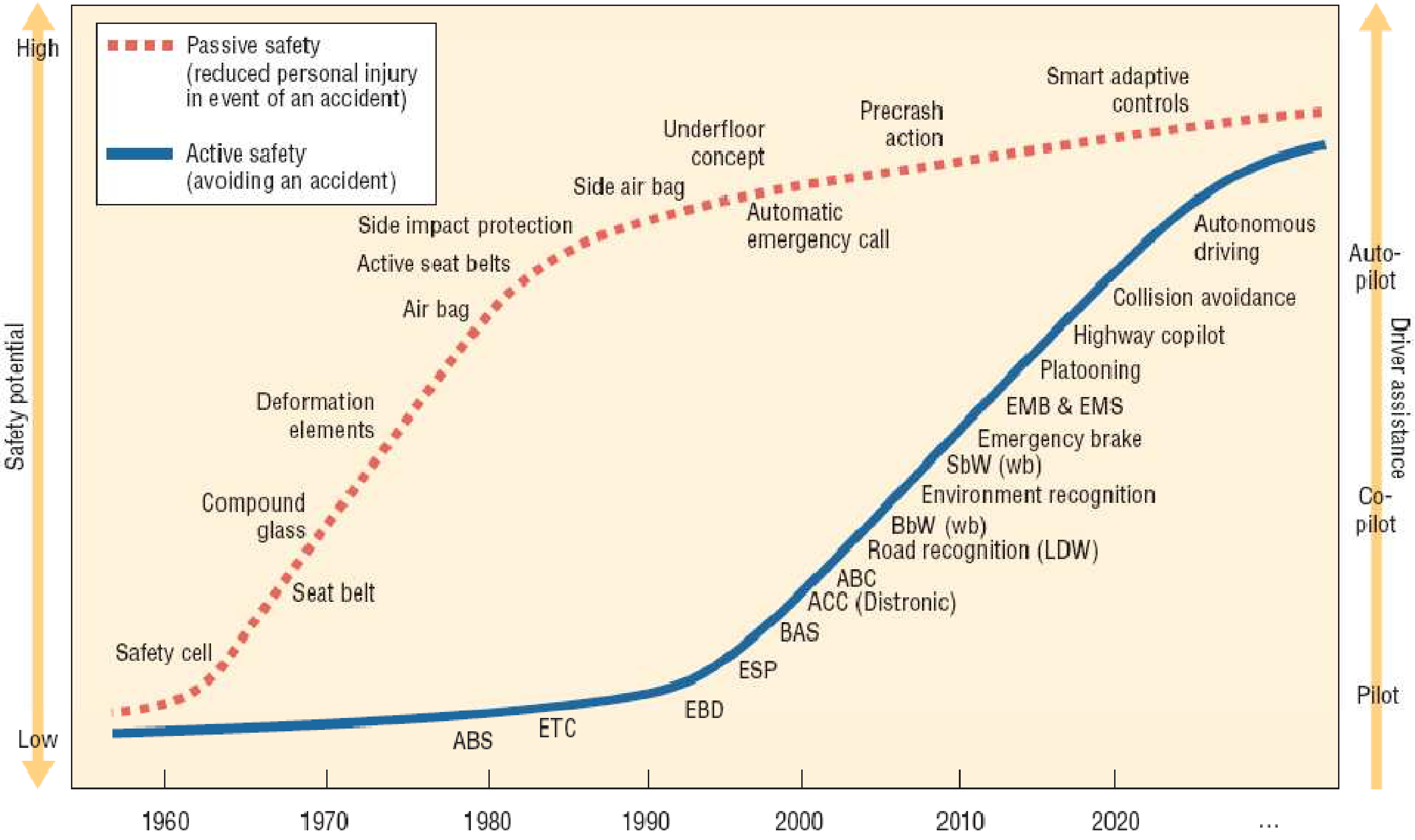
Abhijit Davare, Marco Di Natale, Paolo Giusto, Claudio Pinello, Alberto Sangiovanni-Vincentelli, Wei Zheng, Qi Zhu,

GM Research and Development
University of California at Berkeley,
Scuola Superiore S. Anna,

Outline

- Automotive architecture trends and challenges
- Platform-based system-level design and timing evaluation metrics
- Issues with model-based design
- From analysis to synthesis
- Activation models and end-to-end latencies
- Problem definition
  - Example
- MILP Optimization
- Case Study

Active and Passive Safety



ABC	Active body control	EBD	Electronic brakeforce distribution
ABS	Antilock brake system	EMB	Electromechanical brakes
ACC	Adaptive cruise control	EMS	Electromechanical steering
BAS	Brake assist system	ESP	Electronic stability program
BbW	Brake by wire	ETC	Electronic traction control
CA	Collision avoidance	SbW	Steer by wire
DbW	Drive by wire	(wb)	with mechanical backup

by Leen and Effernan – IEEE Computer

AS - ACC (from Continental web site)

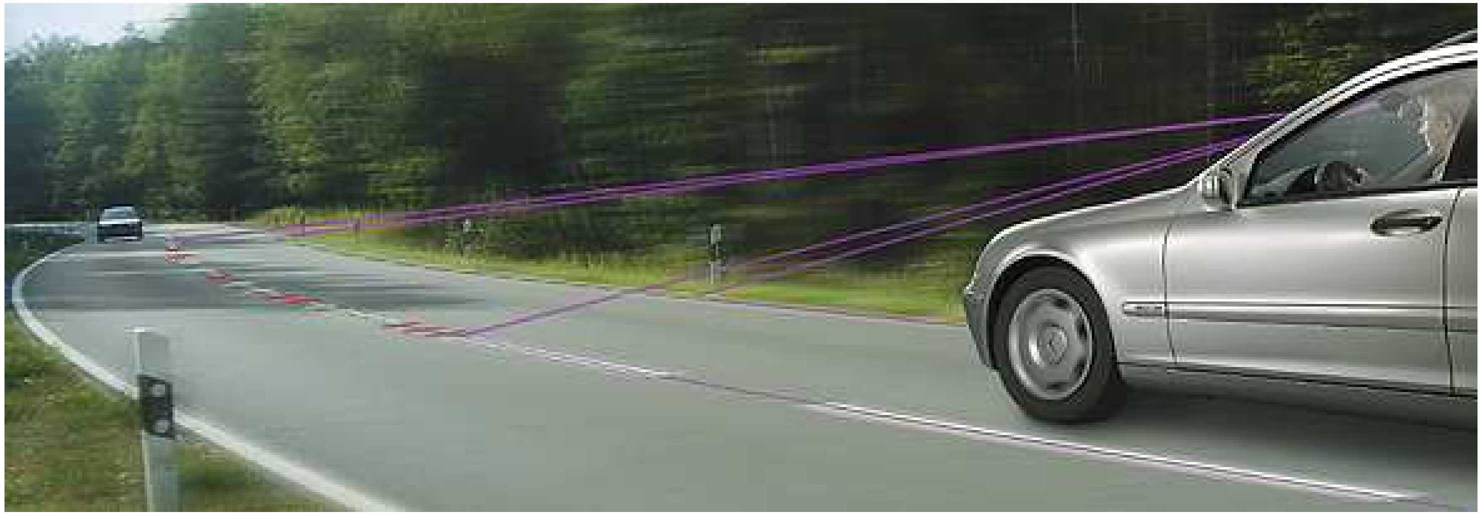
- Adaptive Cruise Control (ACC) – Chassis Electronics Combined with Safety Aspects

As with conventional cruise control, the driver specifies the desired velocity - ACC consistently maintains this desired speed.

In addition, the driver can enter the desired distance to a vehicle driving in front.
If the vehicle now approaches a car travelling more slowly in the same lane, ACC will recognize the diminishing distance and reduce the speed through intervention in the motor management and by braking with a maximum of 0.2 to 0.3 g until the preselected distance is reached. If the lane is clear again, ACC will accelerate to the previously selected desired tempo.

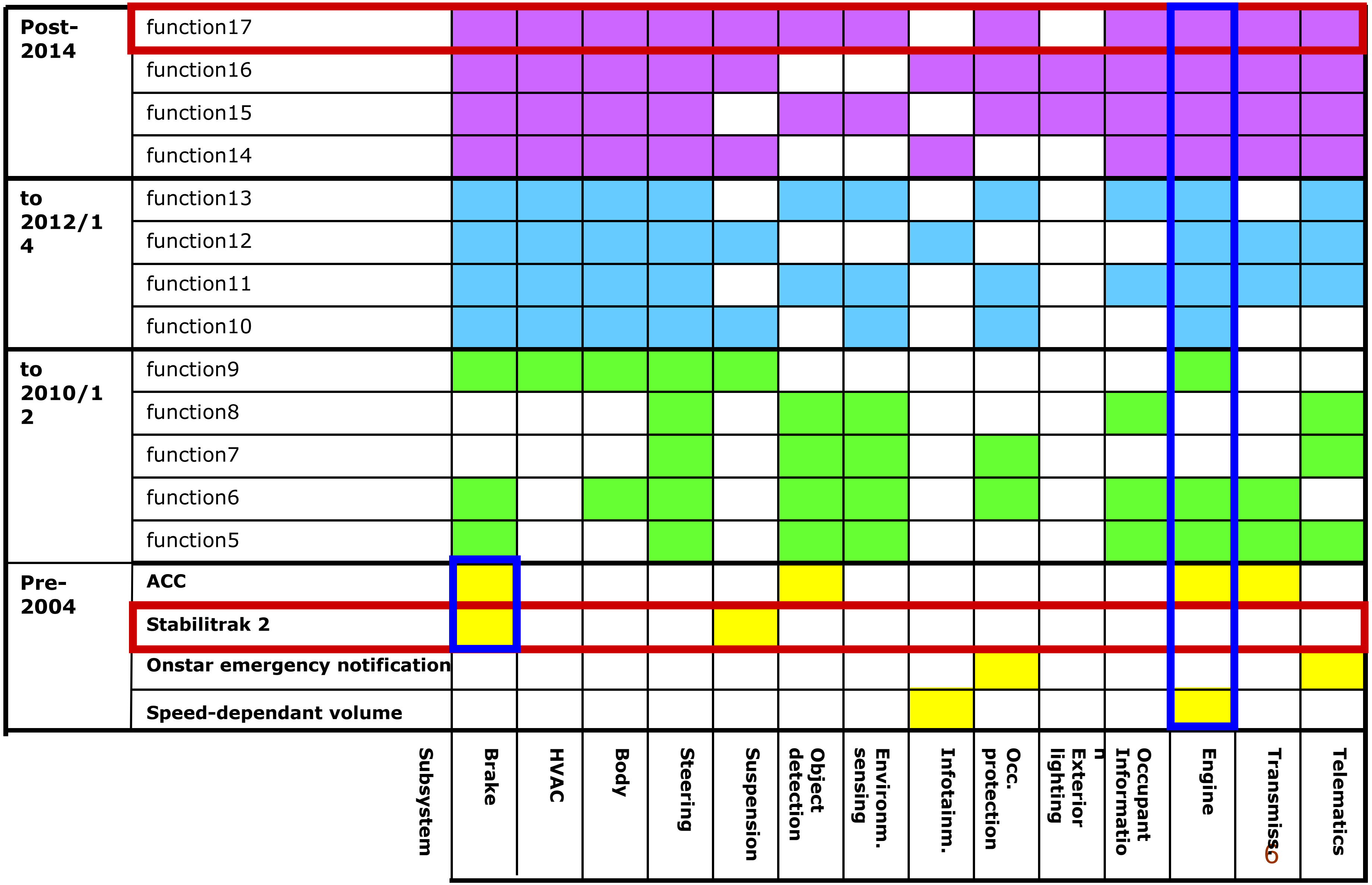
AS-LDW (from Continental web site)

- Lane Departure Warning System (LDW)



LDW wil warn the driver if he or she is on the verge of inadvertently drifting out of the lane. Using a CMOS Camera and an image processing algorithm, this driver assistance system registers the course of the lane in relation to the vehicle. The system "sees", as it were, the course of the road and where the car is going. If the warning algorithm detects an imminent leaving of the current driving lane, the system warns the driver with haptic, kinestatic, or acoustical feedback. Possible warning alerts can be a trembling in the steering wheel, a vibrating seat or a virtual washboard sound. Series production is planned for 2005.

Evolution of Integrated Functions



	Subsystem	Brake	HVAC	Body	Steering	Suspension	Object detection	Environm. sensing	Infotainm.	Occ. protection	Exterior lighting	Occupant Information	Engine	Transmiss	Telematics
Post-2014	function17	🟪	🟪	🟪	🟪	🟪	🟪	🟪		🟪		🟪	🟪	🟪	🟪
	function16	🟪	🟪	🟪	🟪	🟪			🟪	🟪	🟪	🟪	🟪	🟪	🟪
	function15	🟪	🟪	🟪	🟪		🟪	🟪		🟪	🟪	🟪	🟪	🟪	🟪
	function14	🟪	🟪	🟪	🟪	🟪			🟪			🟪	🟪	🟪	🟪
to 2012/14	function13	🟦	🟦	🟦	🟦		🟦	🟦		🟦		🟦	🟦		🟦
	function12	🟦	🟦	🟦	🟦	🟦			🟦				🟦	🟦	🟦
	function11	🟦	🟦	🟦	🟦		🟦	🟦		🟦		🟦	🟦	🟦	🟦
	function10	🟦	🟦	🟦	🟦	🟦		🟦		🟦			🟦		
to 2010/12	function9	🟩	🟩	🟩	🟩	🟩							🟩		
	function8				🟩		🟩	🟩				🟩			🟩
	function7				🟩		🟩	🟩		🟩					🟩
	function6	🟩		🟩	🟩		🟩	🟩		🟩		🟩	🟩	🟩	
	function5	🟩			🟩		🟩	🟩				🟩	🟩	🟩	🟩
Pre-2004	**ACC**	🟨					🟨						🟨	🟨	
	Stabilitrak 2	🟨				🟨									
	Onstar emergency notification									🟨					🟨
	Speed-dependant volume								🟨				🟨		

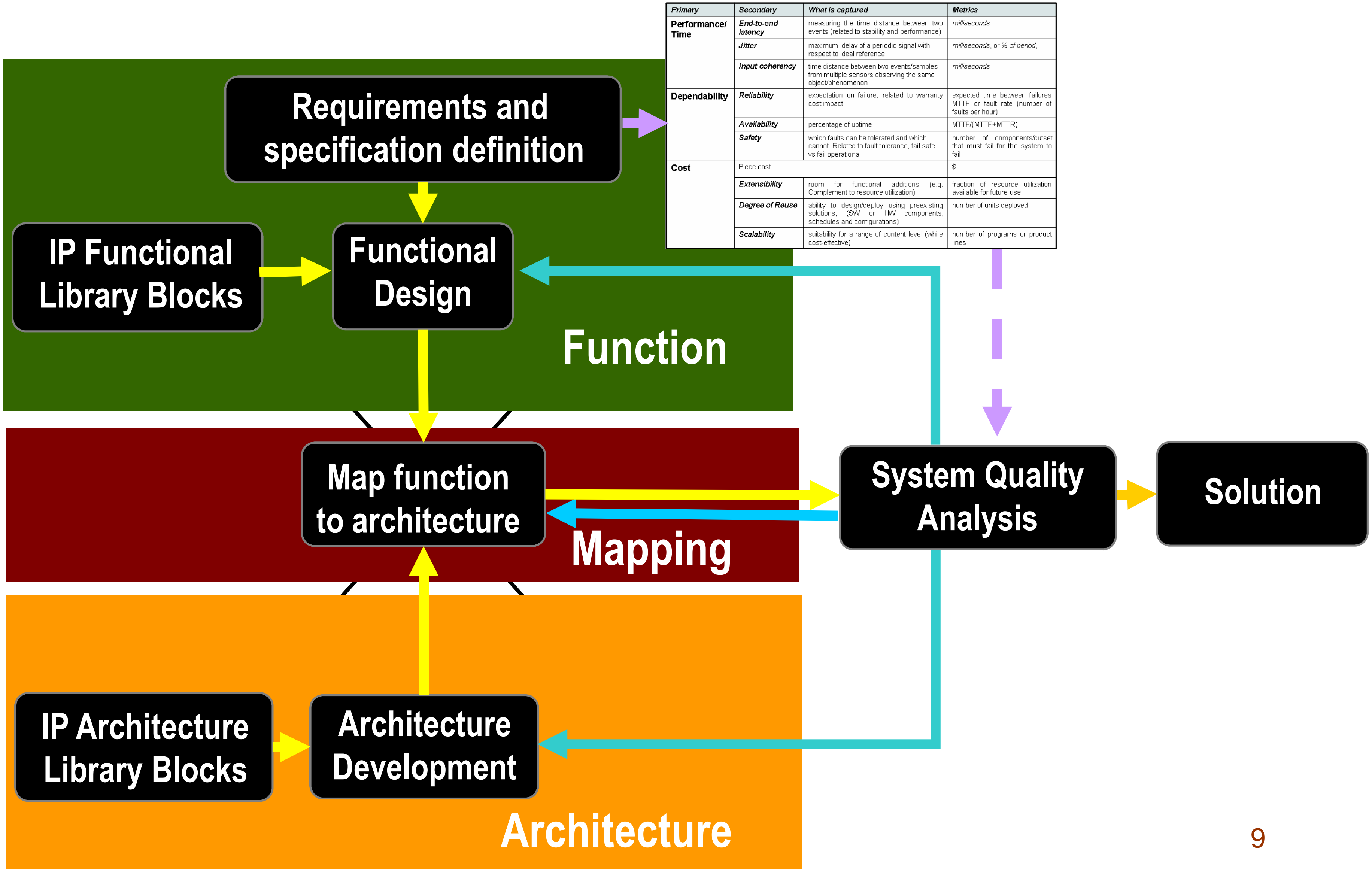
Automotive architecture trends

- Horizontally-integrated functions are becoming key differentiators and are gaining increasing authority
- An increasing number of functions will be distributed on a decreasing number of ECUs and enabled through an increasing number of smart sensors and actuators
  - today: > 5 buses and > 30 ECUs
- 90% of innovation in cars for the foreseeable future will be enabled through the Electronic Vehicle Architecture
- Transition from single-ECU Black-box based development processes to a system-level engineering process
  - System-level methodologies for quantitative exploration and selection,
  - From Hardware Emulation to Model Based Verification of the System
- Architectures need to be defined years ahead of production time, with incomplete information about (future) features
- Multiple non-functional requirements can be defined

Outline

- Automotive architecture trends and challenges
- Platform-based system-level design and timing evaluation metrics
  - worst-case analysis
  - stochastic analysis
- Issues with model-based design
- From analysis to synthesis
- Activation models and end-to-end latencies
- Problem definition
  - Example
- MILP Optimization
- Case Study

Deployment Design Process



Requirements and specification definition

IP Functional Library Blocks

Functional Design

Function

Map function to architecture

Mapping

System Quality Analysis

Solution

IP Architecture Library Blocks

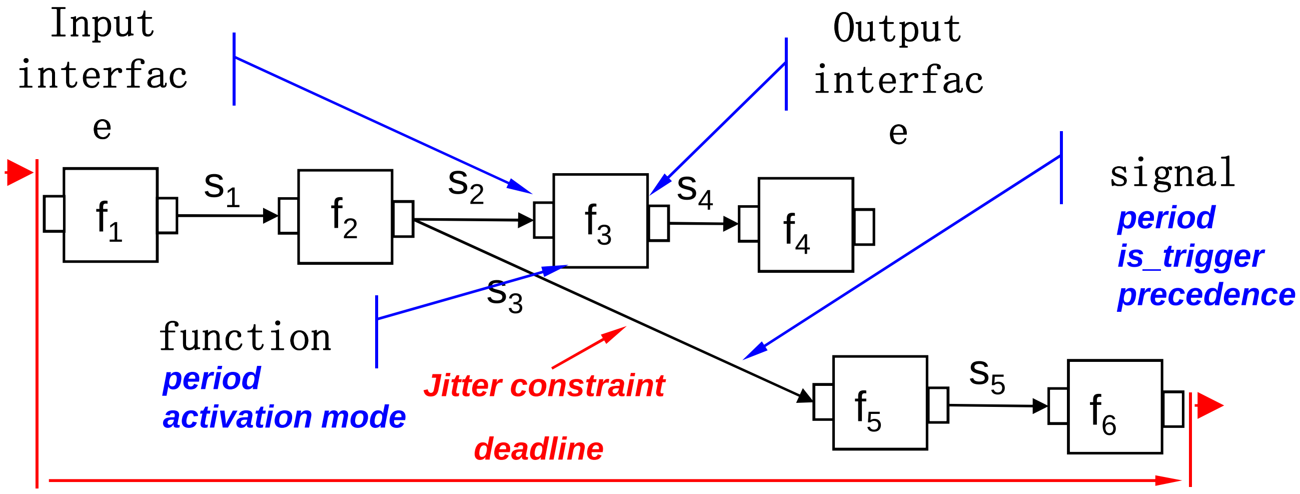
Architecture Development

Architecture

Primary	*Secondary*	*What is captured*	*Metrics*
Performance/ Time	***End-to-end latency***	measuring the time distance between two events (related to stability and performance)	*milliseconds*
	Jitter	maximum delay of a periodic signal with respect to ideal reference	*milliseconds*, or *% of period*,
	Input coherency	time distance between two events/samples from multiple sensors observing the same object/phenomenon	*milliseconds*
Dependability	***Reliability***	expectation on failure, related to warranty cost impact	expected time between failures MTTF or fault rate (number of faults per hour)
	Availability	percentage of uptime	MTTF/(MTTF+MTTR)
	Safety	which faults can be tolerated and which cannot. Related to fault tolerance, fail safe vs fail operational	number of components/cutset that must fail for the system to fail
Cost	Piece cost		$
	Extensibility	room for functional additions (e.g. Complement to resource utilization)	fraction of resource utilization available for future use
	Degree of Reuse	ability to design/deploy using preexisting solutions, (SW or HW components, schedules and configurations)	number of units deployed
	Scalability	suitability for a range of content level (while cost-effective)	number of programs or product lines

Functional model

Functional model



Input interface

Output interface

signal

period

is_trigger

precedence

function

period

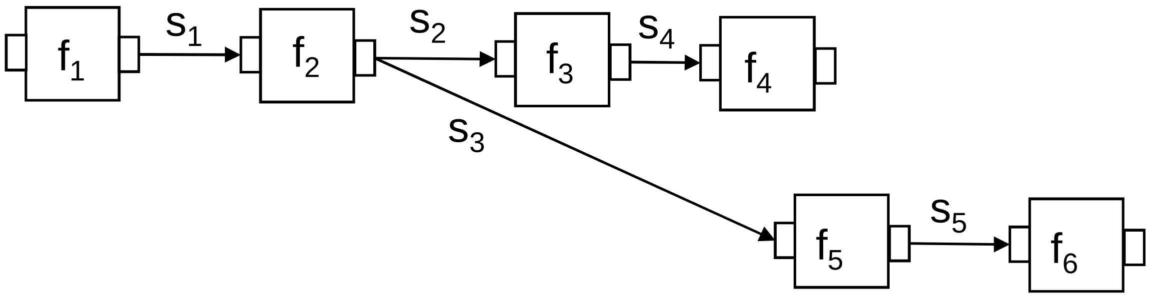
activation mode

Jitter constraint

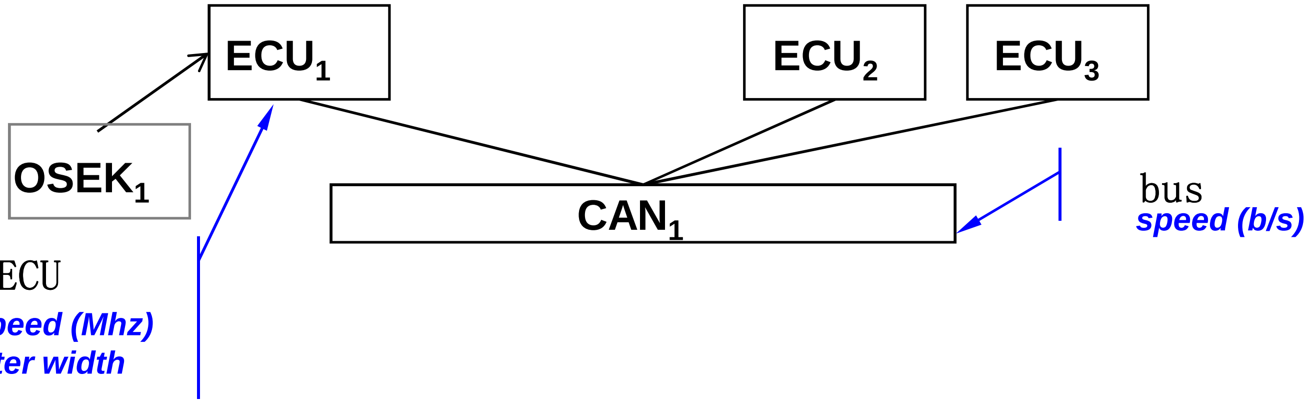
deadline

Architecture model

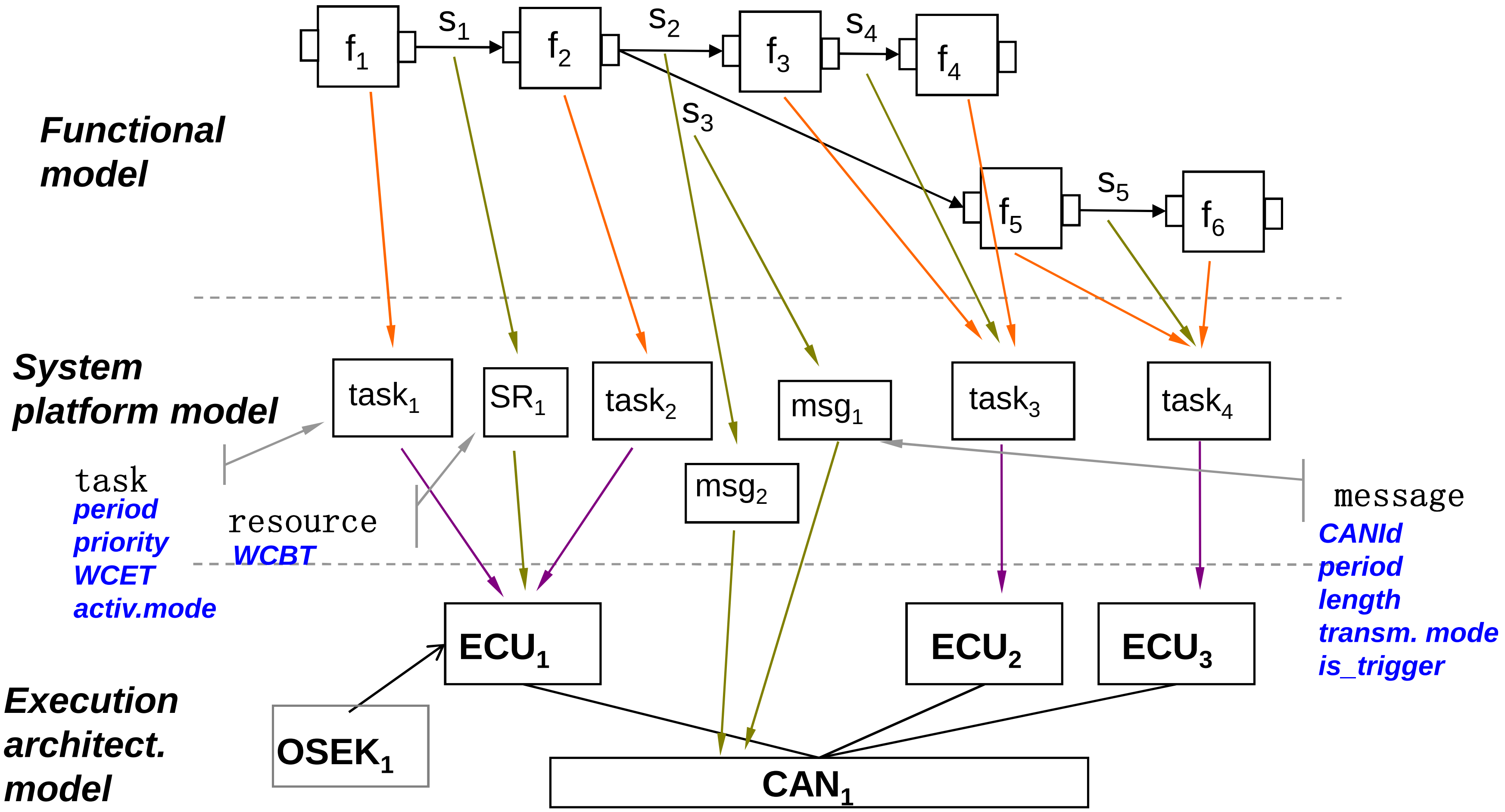
Functional model



Execution architect. model



Deployment model



Functional model

System platform model

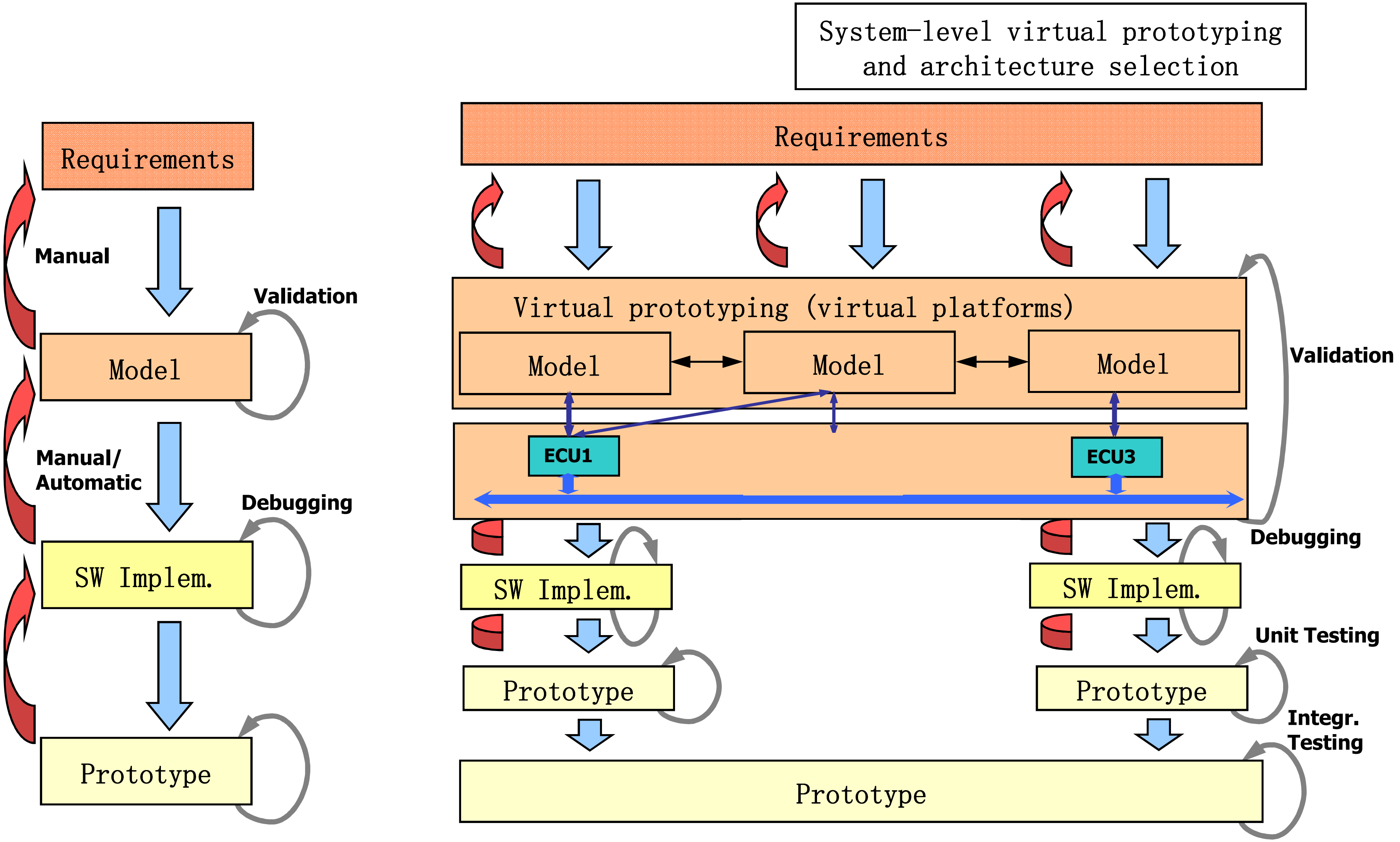
Execution architect. model

task
period
priority
WCET
activ.mode

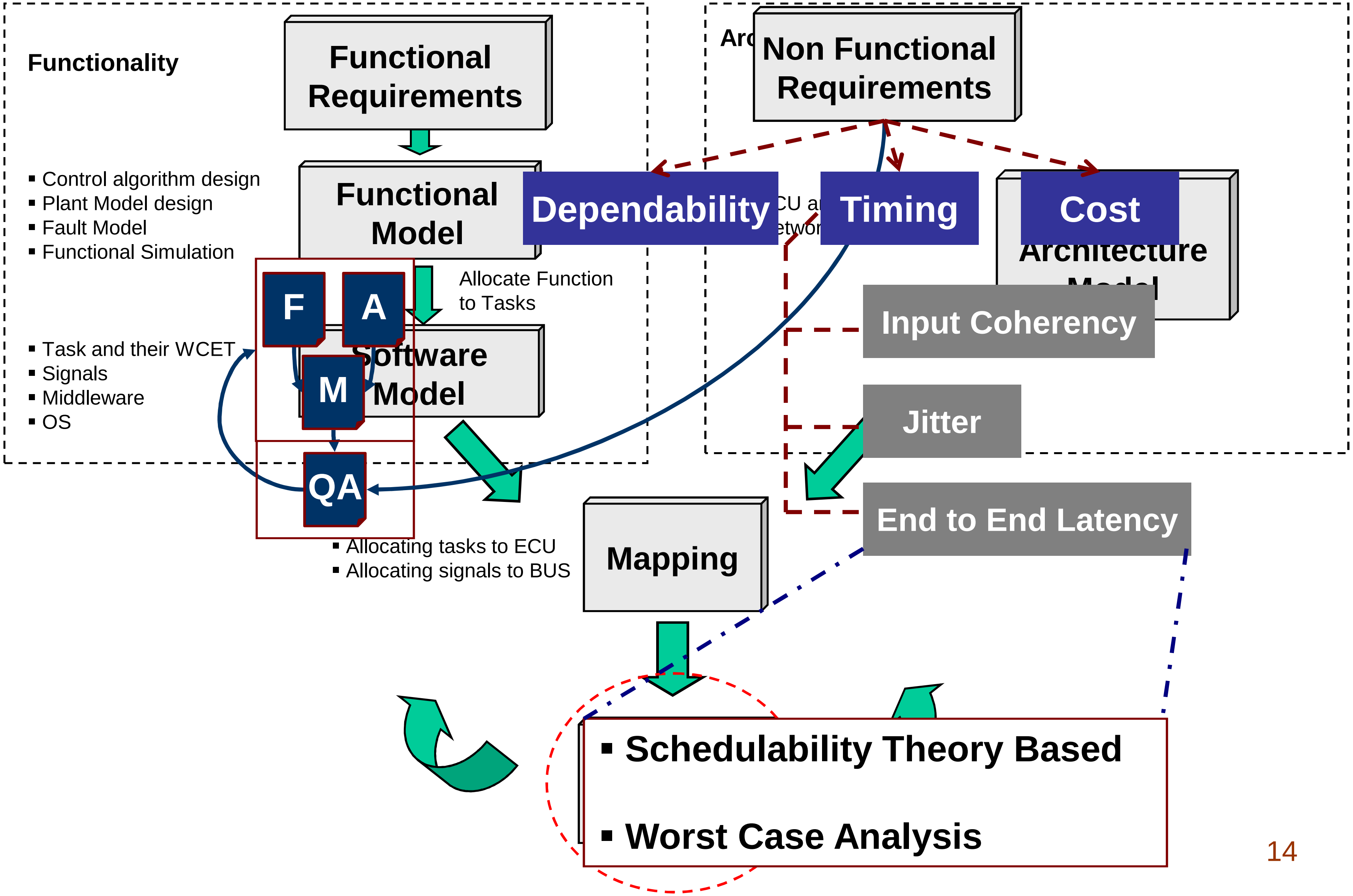
resource
WCBT

message
CANId
period
length
transm. mode
is_trigger

Tool integration platform



Design Process and Requirement



Functionality

Functional Requirements

- Control algorithm design
- Plant Model design
- Fault Model
- Functional Simulation

Functional Model

Allocate Function to Tasks

- Task and their WCET
- Signals
- Middleware
- OS

Software Model

F **A** **M** **QA**

- Allocating tasks to ECU
- Allocating signals to BUS

Arc

Non Functional Requirements

CU a
etwor

Dependability **Timing** **Cost**

Architecture

Input Coherency

Jitter

End to End Latency

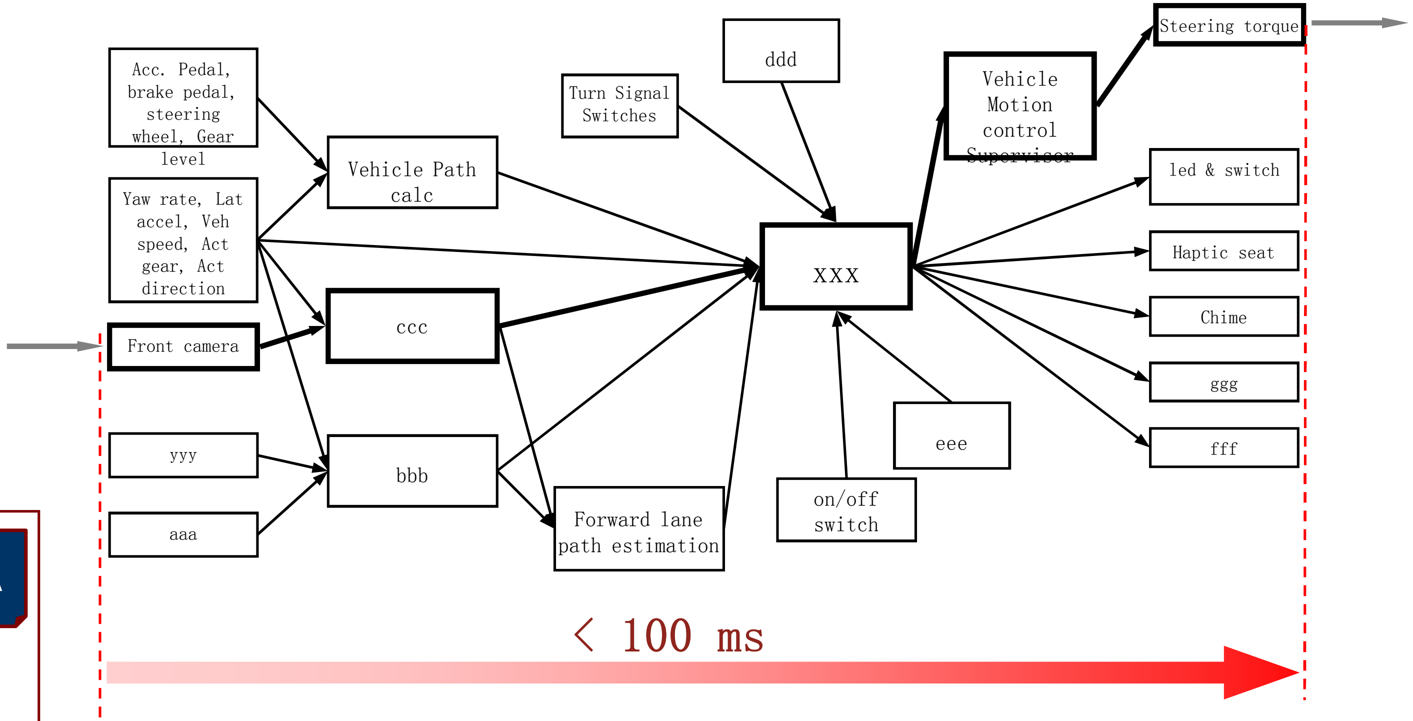
Mapping

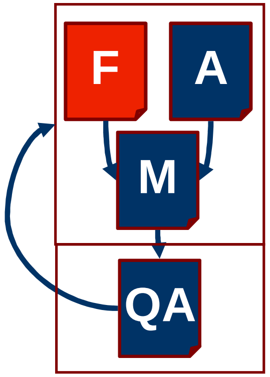
- **Schedulability Theory Based**
- **Worst Case Analysis**

Functional Model: An example

Function Example
xxx

Functional Design



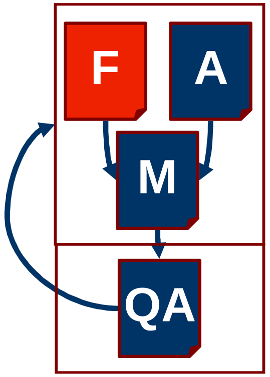


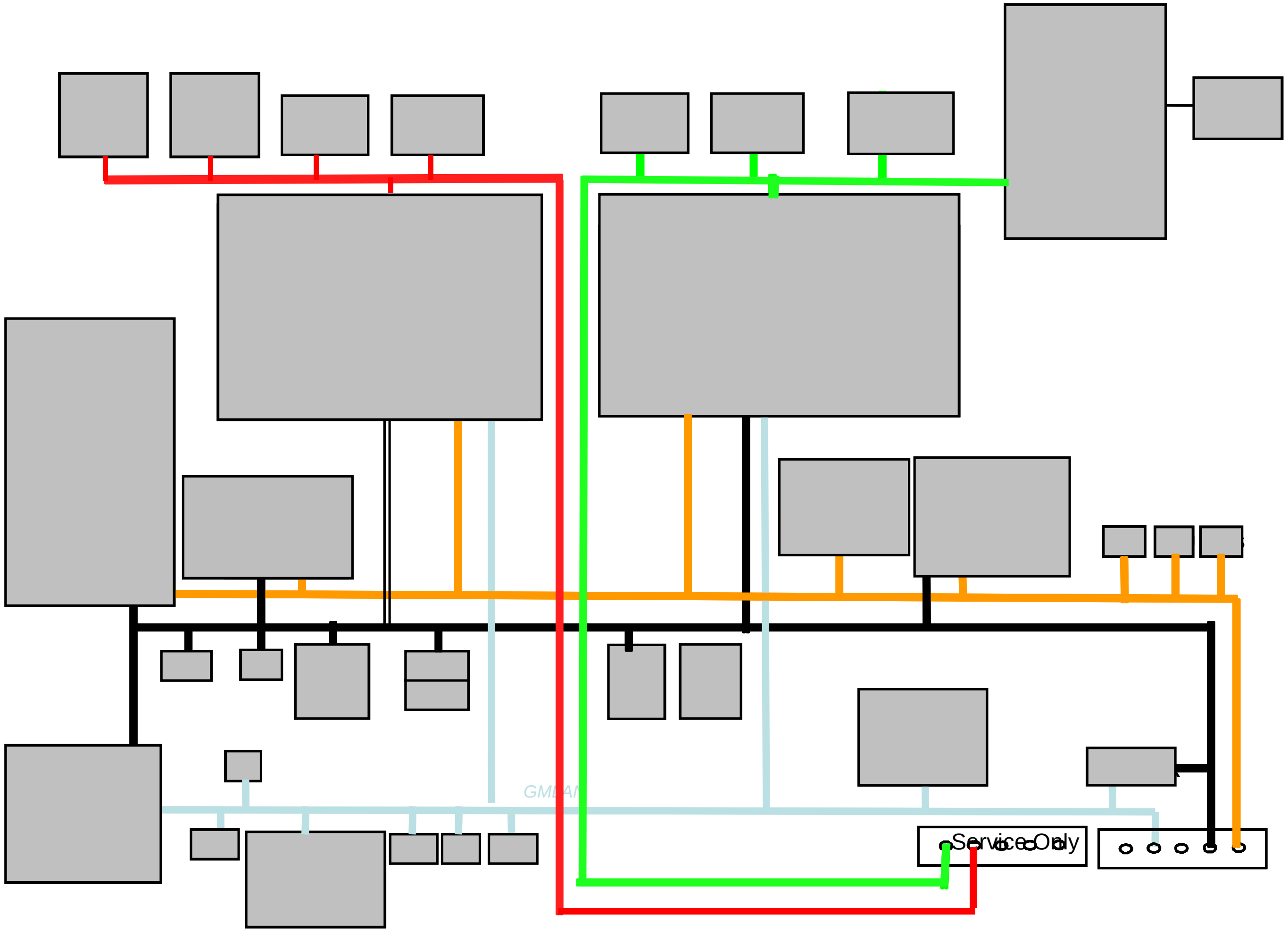
Requirements and specification definition

Architecture Model: An example

Architecture
Option

Architecture Development

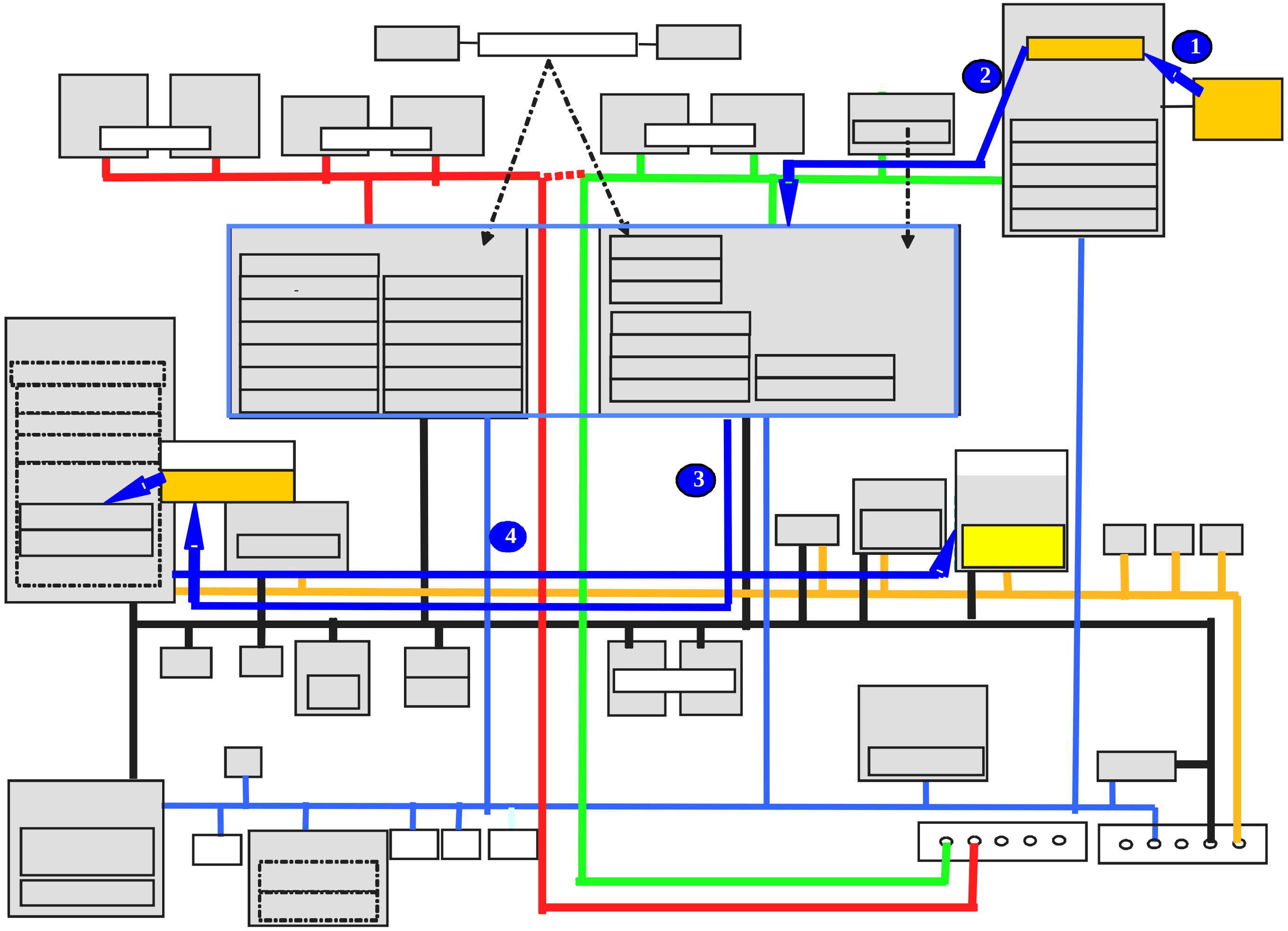


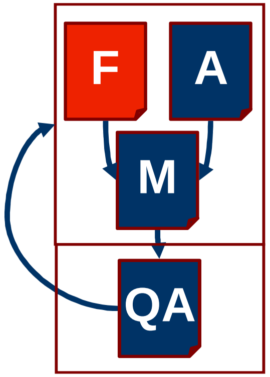


Deployment: An example

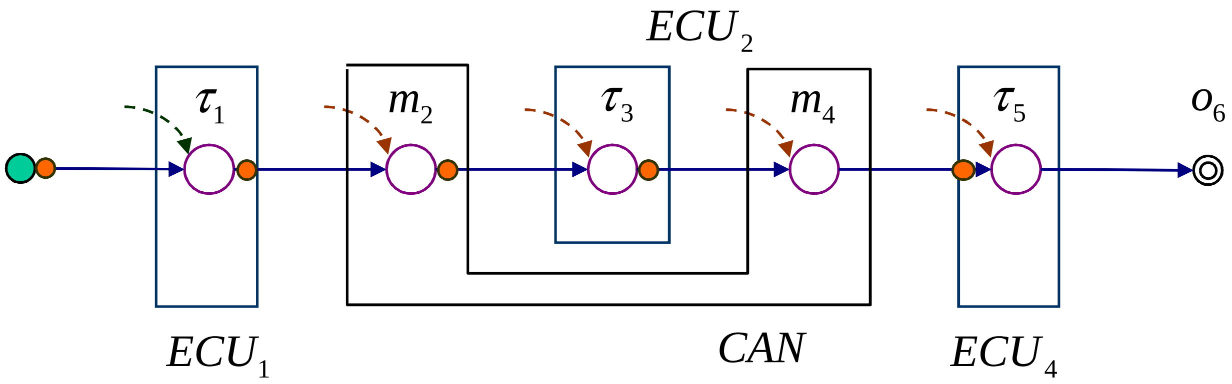
End-to-end latencies

ECU and bus utilizations

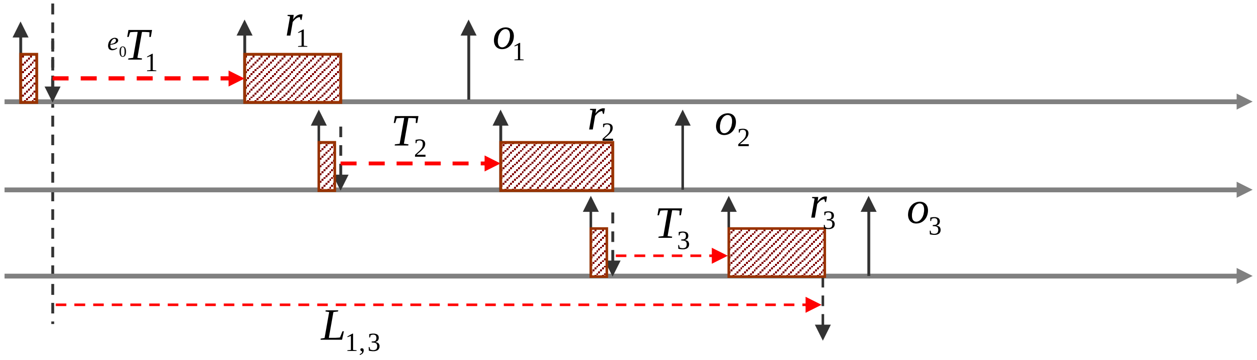




Periodic Activation Model



- Predictable activation model easy latency computation
- Suffers from high worst case latencies



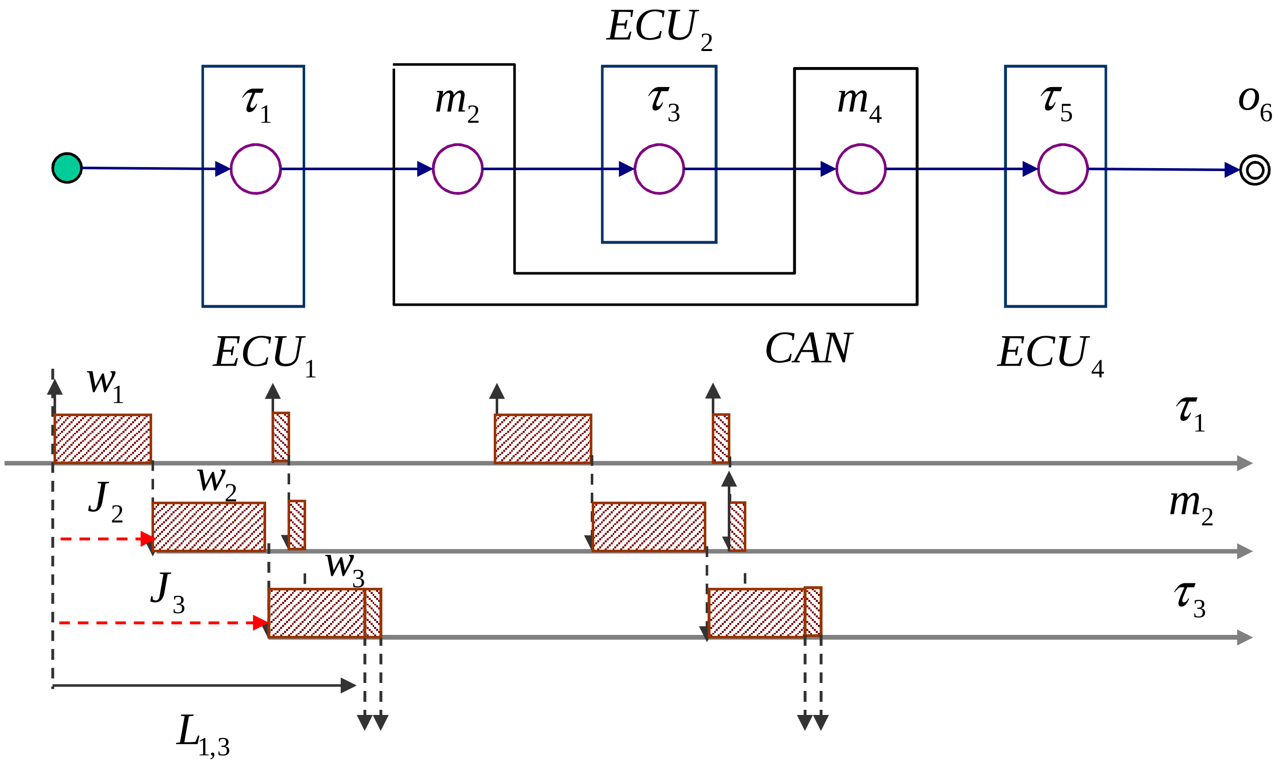
$$L_{i,j} = \sum_{k:o_k \in P(i,j)} (T_k + r_k)$$

Where

$$r_i = C_i + \sum_{j \in hp(i)} \left\lceil \frac{r_i}{T_j} \right\rceil C_j$$

$$L_{1,3} = T_1 + r_1 + T_2 + r_2 + T_3 + r_3$$

Data Driven Activation Model



- Shorter end to end latencies
- Large interference intervals with bursty activations

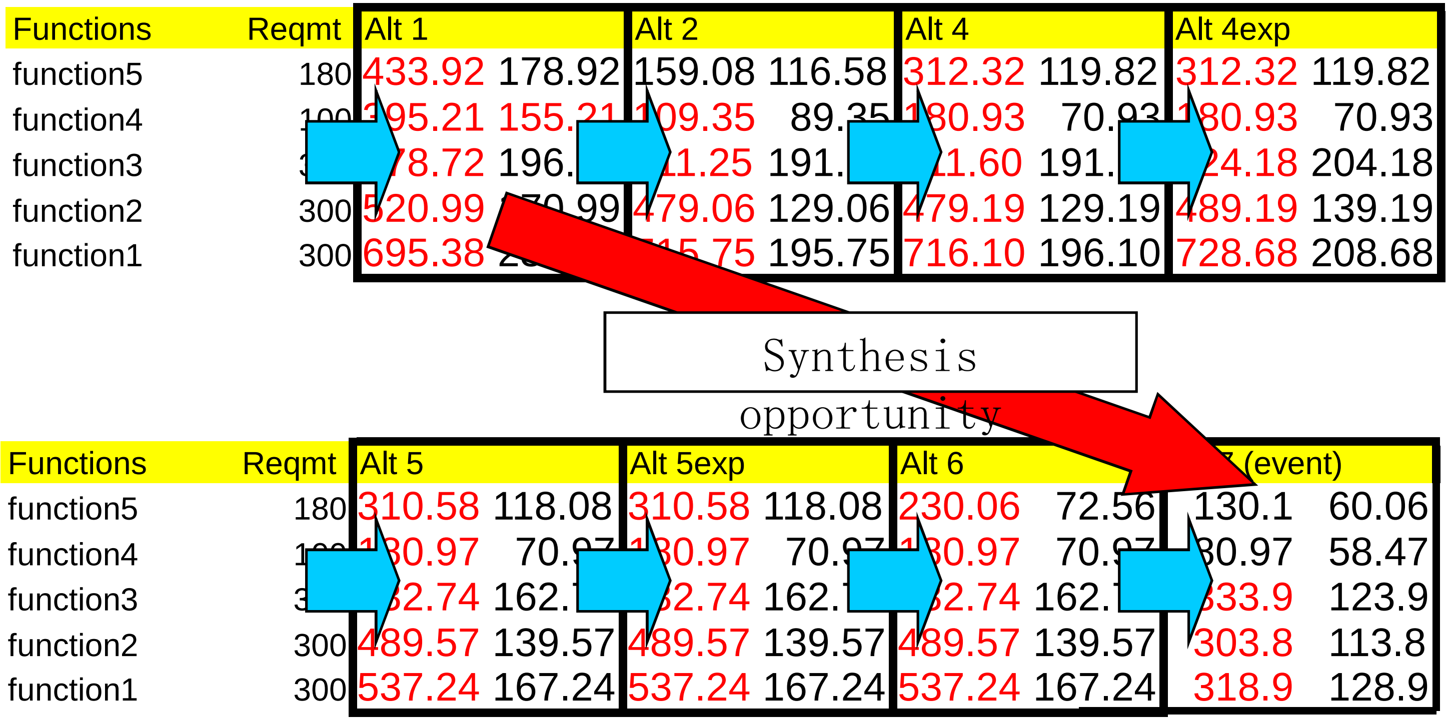
$$L_{i,j} = \sum_{k:o_k \in P(i,j)} w_k$$

Where Approx.

$$w_i = C_i + \sum_{j \in hp(i)} \left\lceil \frac{w_i + J_j}{T_j} \right\rceil C_j$$

$$L_{1,3} = w_1 \quad + w_2 \quad + w_3$$

Case study 1



Functions	Reqmt	Alt 1		Alt 2		Alt 4		Alt 4exp	
function5	180	433.92	178.92	159.08	116.58	312.32	119.82	312.32	119.82
function4	100	395.21	155.21	109.35	89.35	180.93	70.93	180.93	70.93
function3	3	78.72	196.	11.25	191.	11.60	191.	24.18	204.18
function2	300	520.99	170.99	479.06	129.06	479.19	129.19	489.19	139.19
function1	300	695.38	2	715.75	195.75	716.10	196.10	728.68	208.68

Synthesis opportunity

Functions	Reqmt	Alt 5		Alt 5exp		Alt 6		7 (event)	
function5	180	310.58	118.08	310.58	118.08	230.06	72.56	130.1	60.06
function4	100	180.97	70.97	180.97	70.97	180.97	70.97	80.97	58.47
function3	3	32.74	162.	32.74	162.	32.74	162.	33.9	123.9
function2	300	489.57	139.57	489.57	139.57	489.57	139.57	303.8	113.8
function1	300	537.24	167.24	537.24	167.24	537.24	167.24	318.9	128.9

- By transmitting messages “on event”, the worst case latency can be reduced in most cases
- By properly allocating functions to ECUs the end-2-end latency can be improved

Stochastic and simulation-based analysis

- Simulation
  - Built C++ simulator for can message analysis (at bit level – only arbitration)
  - Currently being expanded to end-to-end computations, periodic sampling model for latency analysis
- Stochastic analysis
  - Approximate analysis of pmf of message latencies in CAN bus (complete - target ?)
  - Future work
    - End-to-end analysis of sampling model
    - Regression-based analysis to define pmf from general information (such as load or loads at harmonic rates)

Stochastic and simulation-based analysis

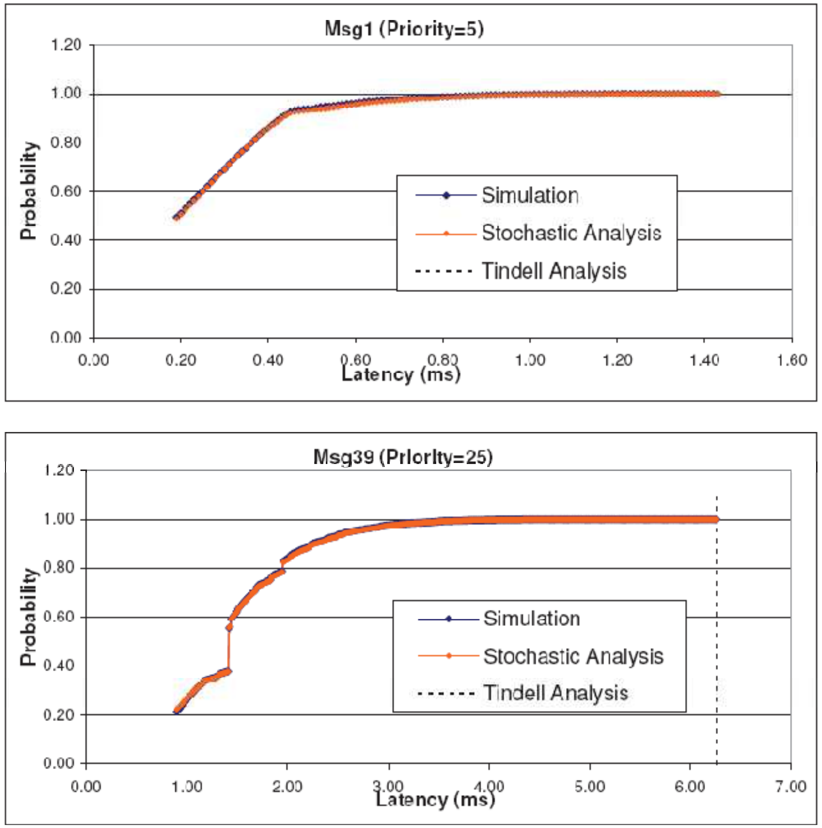


Figure 5. Latency *cdfs* of two high priority representative messages in the test set

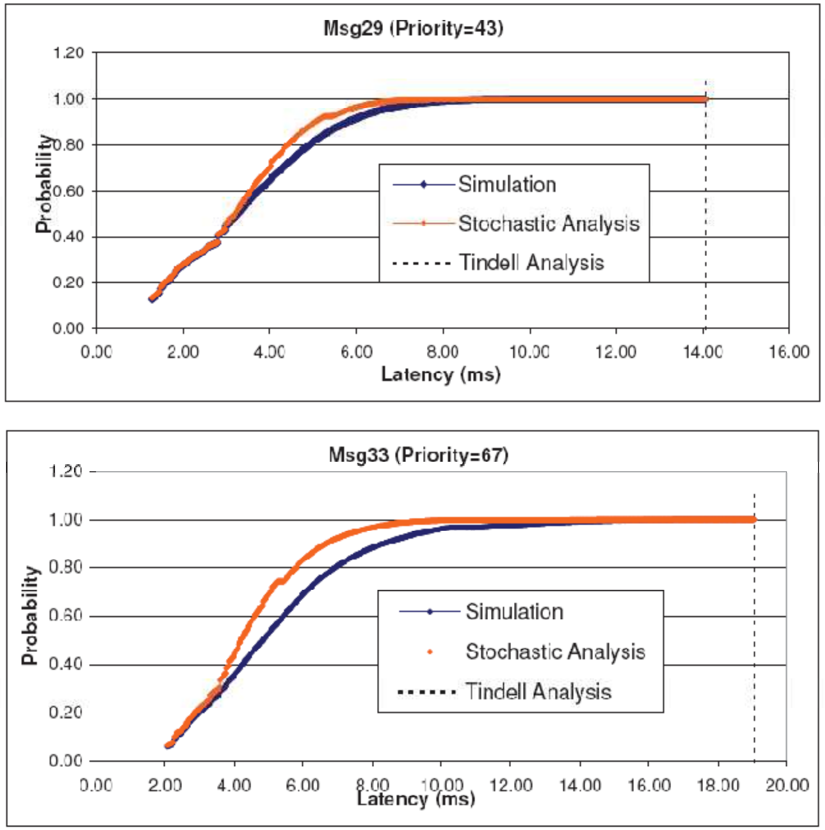


Figure 6. Latency *cdfs* of two low priority representative messages in the test set

62 msg set (subset of chassis bus). Low priority msg – Distributions of latencies

Outline

- Automotive architecture trends and challenges
- Platform-based system-level design and timing evaluation metrics
  - worst-case analysis
  - stochastic analysis
- Issues with model-based design
- From analysis to synthesis
- Activation models and end-to-end latencies
- Problem definition
  - Example
- MILP Optimization
- Case Study

Issues with model-based development

- Model-based design methodologies
  - improve the quality and the reusability of software.
  - The possibility of defining components (subsystems) at higher levels of abstraction and with well defined interfaces allows separation of concerns and improves modularity and reusability.
  - The availability of verification tools (often by simulation) gives the possibility of a design-time verification of the system properties.
- However, most modern tools for model-based design have a number of shortcomings

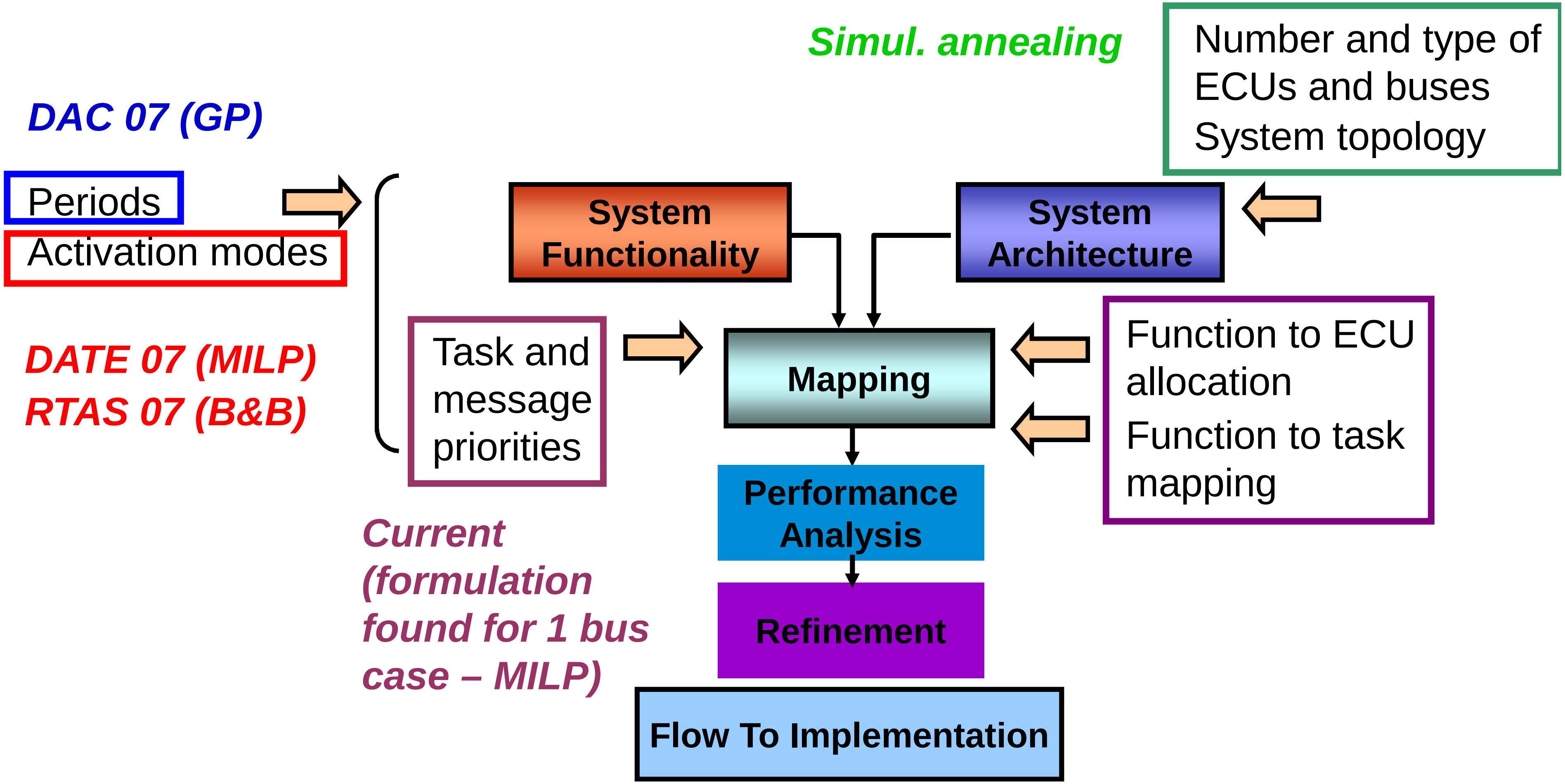
Issues with model-based development

- *Lack of separation between the functional model and the architecture model*
- *Lack of support for the definition of the task and resource model*
- *Insufficient support for the specification of timing constraints and attributes*
- *Lack of modeling support for the analysis and the back-annotation of scheduling-related delays*
- *Issue of semantics preservation*

Outline

- Automotive architecture trends and challenges
- Platform-based system-level design and timing evaluation metrics
  - worst-case analysis
  - stochastic analysis
- Issues with model-based design
- Time predictability and timing isolation
- From analysis to synthesis
- Activation models and end-to-end latencies
- Problem definition
  - Example
- MILP Optimization
- Case Study

Opportunities for synthesis



DAC 07 (GP)

Periods

Activation modes

DATE 07 (MILP)

RTAS 07 (B&B)

Task and message priorities

Current (formulation found for 1 bus case – MILP)

Simul. annealing

Number and type of ECUs and buses

System topology

System Functionality

System Architecture

Mapping

Function to ECU allocation

Function to task mapping

Performance Analysis

Refinement

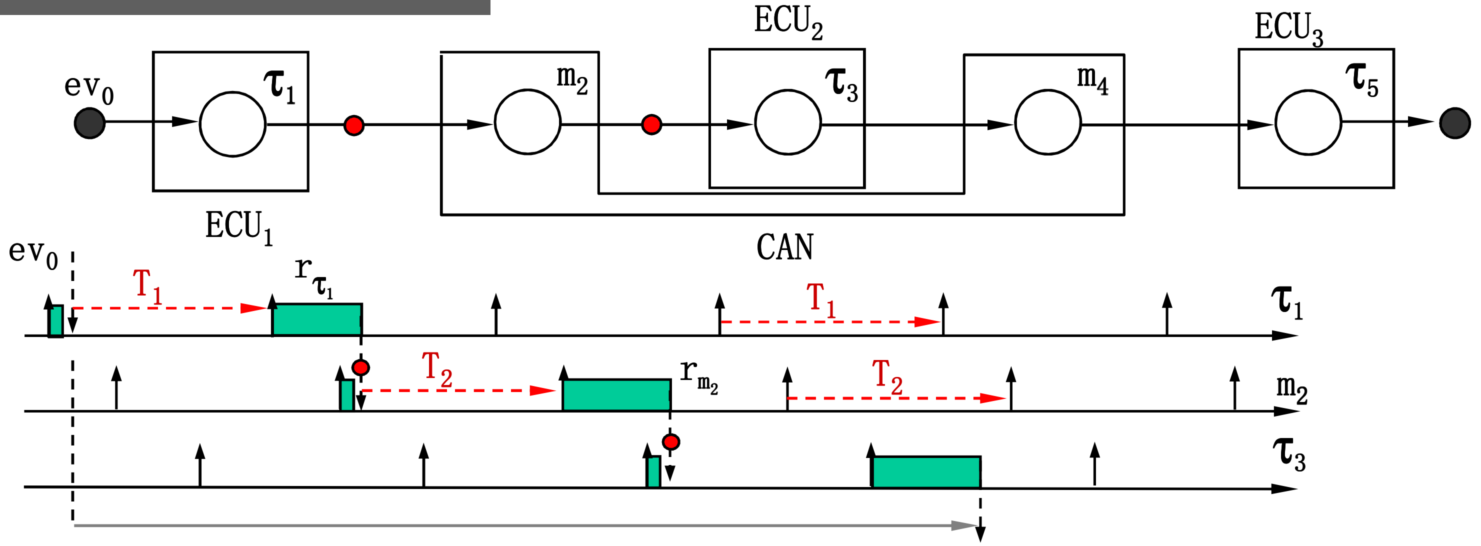
Flow To Implementation

Periodic Activation Model

High latency, but allows decoupling the scheduling problem

End-to-end latency analysis

Periodic asynchronous activation model



$$l_{(i,j)} = \sum_{k:o_k \in P(i,j)} (T_k + r_k)$$

where (approx.)

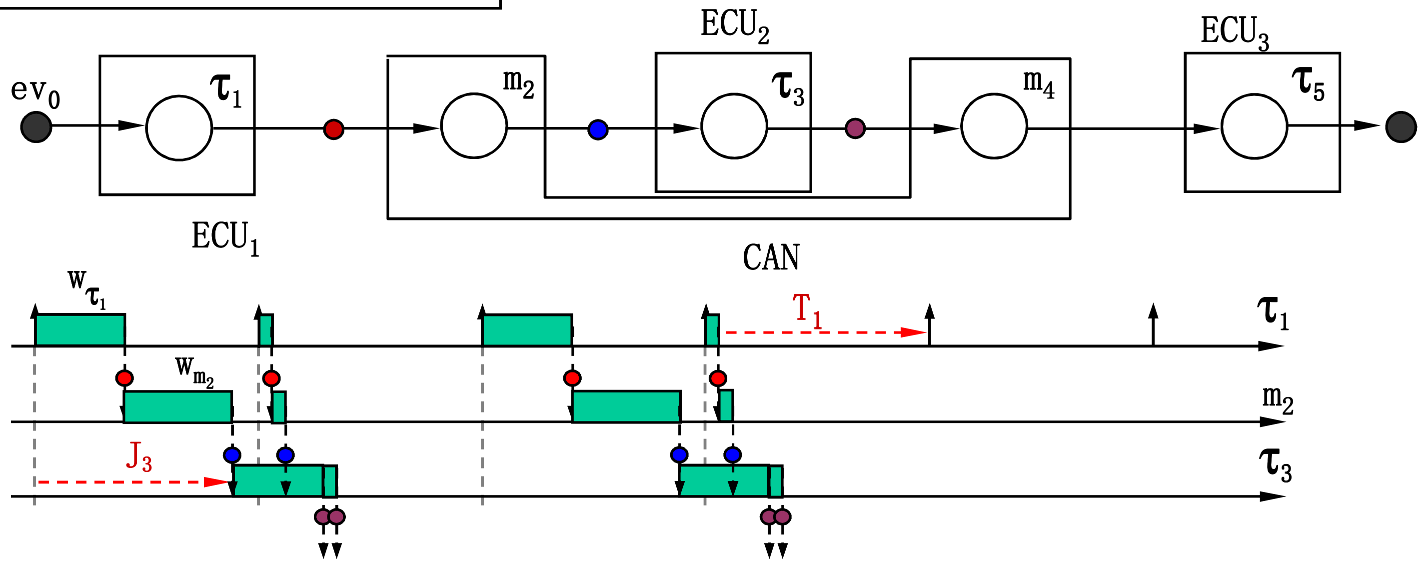
$$r_i = C_i + \sum_{j \in hp(i)} \left\lceil \frac{r_i}{T_j} \right\rceil C_j$$

Event-based Activation Model

Lower latency for high priority paths, jitter increases along the path

End-to-end latency analysis

Data-driven precedence constrained activation model

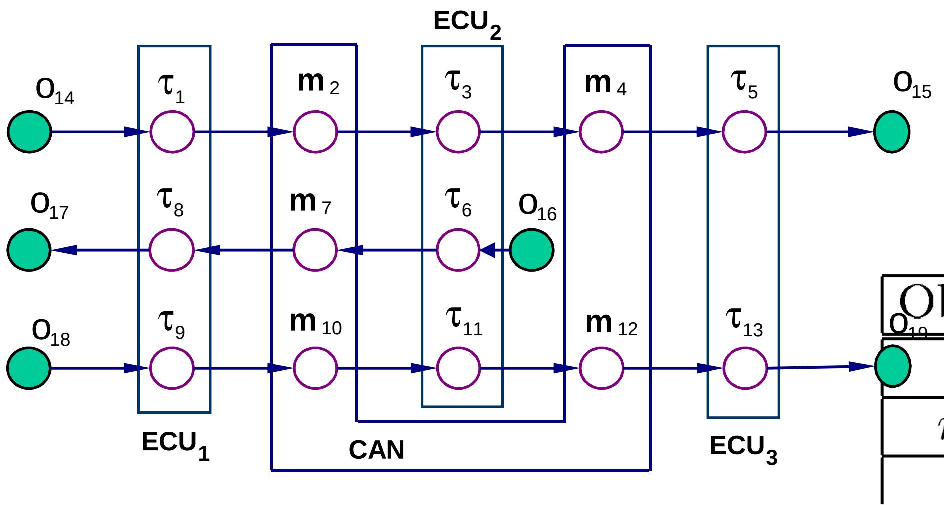


$$l_{(i,j)} = \sum_{k:o_k \in P(i,j)} w_k$$

where (approx.)

$$w_i = C_i + \sum_{j \in hp(i)} \left\lceil \frac{w_i + J_j}{T_j} \right\rceil C_j$$

Activation modes: latency tradeoffs



Periodic

Event-based

Object	π_i	T_i	C_i	r_i	l_i	J_i	w_i	r_i
τ_1	13	15	8	8	8	0	8	8
m_2	12	15	2	4	27	8	4	12
τ_3	11	15	8	8	50	12	8	20
m_4	10	15	2	6	71	20	6	26
τ_5	9	15	6	6	~~92~~	26	6	**32**
τ_6	8	40	6	14	14	0	30	30
m_7	7	40	2	8	62	30	12	42
τ_8	6	40	14	30	~~132~~	42	30	**72**
τ_9	5	30	2	42	42	0	190	190
m_{10}	4	30	2	10	82	190	18	208
τ_{11}	3	30	6	28	140	208	58	266
m_{12}	2	30	2	10	180	266	36	302
τ_{13}	1	30	8	14	~~224~~	302	32	~~334~~

End to end latency requirements

$d_{o_{14},o_{15}} \Rightarrow$ **70**

$d_{o_{16},o_{17}} \Rightarrow$ **100**

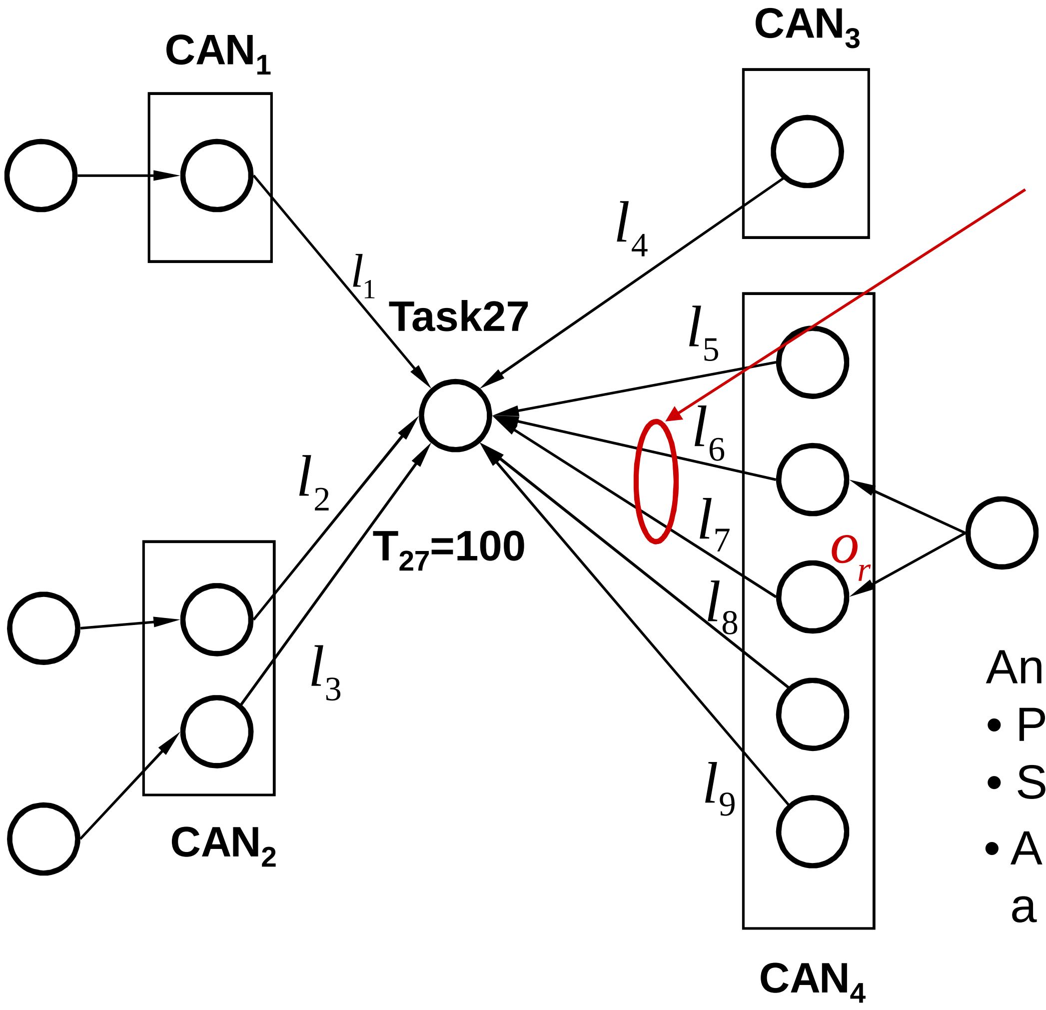
$d_{o_{18},o_{19}} \Rightarrow$ **120**

Mixed activation mode

$$L_{i,j} = \sum_{k:k=j \vee l_{k,q} \in \mathcal{E}_p} (J_k + w_k) + \sum_{q:l_{k,q} \in \mathcal{E}_p} T_q$$

Model Definition

- Selection of the activation event and link groups



Link Group

Group Representative

$$o_r = rep(\mathrm{lg}_1)$$

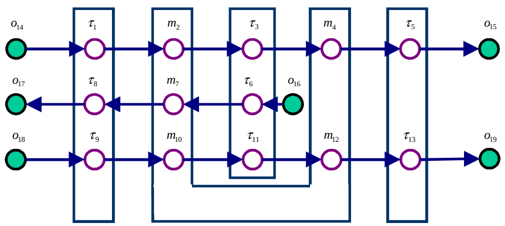
An object can be activated by:

- Periodic Timer
- Signal from a single predecessor
- AND composition of signals from a link group

Latencies of OSEK Tasks and CAN Messages

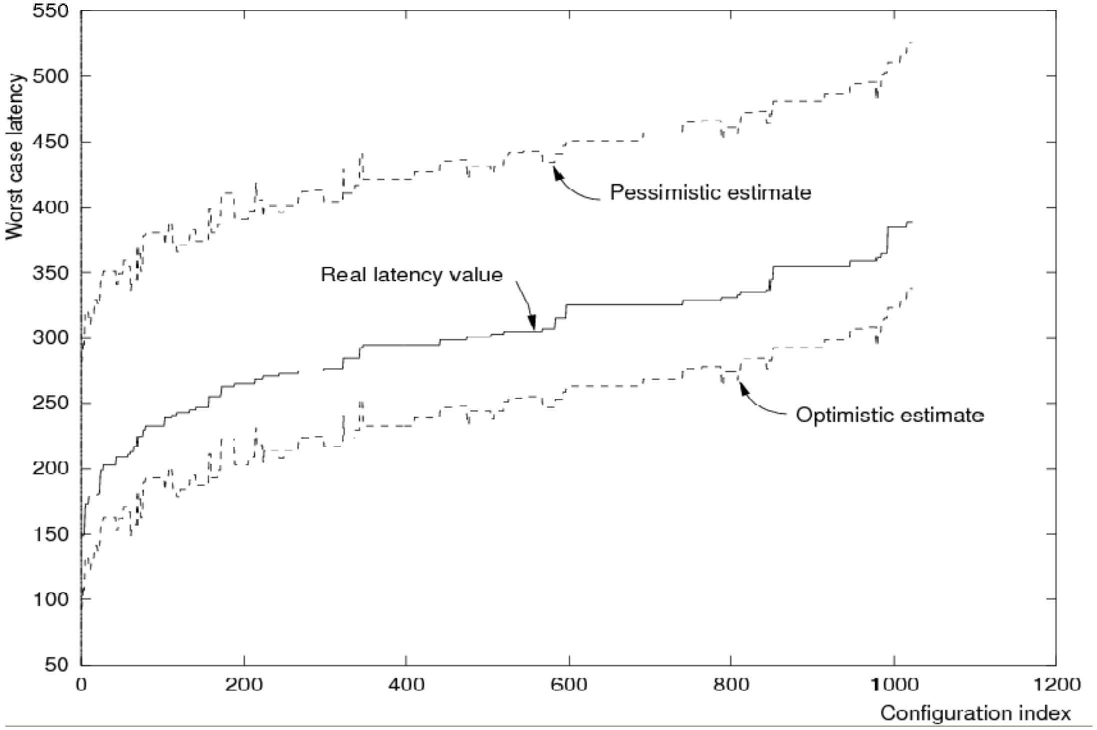
		Linear Combination	First Instance
Processor	Upper	$w_i^{\uparrow} = C_i + \sum_{j\in hp(i)} (\frac{w_i^{\uparrow} + J_j}{T_j} + 1)C_j$ $w_i(\alpha) = \frac{C_i + \alpha \times \sum_{j\in hp(i)} C_j + \sum_{j\in hp(i)} J_j u_j}{1 - \sum_{j\in hp(i)} u_j}$ $w_i(q) = (q+1)C_i + \sum_{j\in hp(i)} \left\lceil \frac{w_i(q) + J_j}{T_j} \right\rceil C_j$ $w_i = \max_q \{w_i(q) - qT_i\}$ $r_i = J_i + w_i$ $q = 0 \ldots q^*, r_i(q^*) \le T_i$ [Jos78]	$w_i = C_i + \sum_{j\in hp(i)} \left\lceil \frac{w_i + J_j}{T_j} \right\rceil C_j$ $r_i = J_i + w_i$
	Lower	$w_i^{\downarrow} = C_i + \sum_{j\in hp(i)} (\frac{w_i^{\downarrow} + J_j}{T_j})C_j$	
Bus	Upper	$wq_i^{\uparrow} = B_i + \sum_{j\in hp(i)} (\frac{wq_i^{\uparrow} + J_j}{T_j} + 1)C_j$ $wq_i(\alpha) = \frac{B_i + \alpha \times \sum_{j\in hp(i)} C_j + \sum_{j\in hp(i)} J_j u_j}{1 - \sum_{j\in hp(i)} u_j}$ $wq_i(q) = B_i + qC_i + \sum_{j\in hp(i)} \left\lceil \frac{wq_i(q) + J_j}{T_j} \right\rceil C_j$ $w_i = \max_q \{C_i + wq_i(q) - qT_i\}$ $r_i = J_i + w_i$ $q = 0 \ldots q^*, r_i(q^*) \le T_i$	$\boldsymbol{wq_i = B_i + \sum_{j\in hp(i)} \left\lceil \frac{wq_i + J_j}{T_j} \right\rceil C_j}$ $\boldsymbol{w_i = wq_i + C_i}$ $\boldsymbol{r_i = J_i + w_i}$
	Lower	$wq_i^{\downarrow} = B_i + \sum_{j\in hp(i)} (\frac{wq_i^{\downarrow} + J_j}{T_j})C_j$	

Linear Approximation



	$L_{o_{14},o_{15}}$	$L_{o_{16},o_{17}}$	$L_{o_{18},o_{19}}$
Linear_upper	44.36	130.86	507.03
Fixed_point	40	88	312
Linear_lower	38.91	79.43	294.96

A linear combination of linear upper and lower bounds can be sufficiently accurate to be used as an estimator of actual e2e latency



MILP Solution

Sets

V : Set of objects implementing the computation and communication functions

E : Set of links connecting schedulable objects

R : Set of resources (CAN, ECUs)

Parameters

π_i : Priority of object o_i

T_i : Period of object o_i

C_i : Worst case execution/transmission time of object o_i

Variables

r_i : Worst case response time of object o_i

J_i : Release Jitter of object o_i

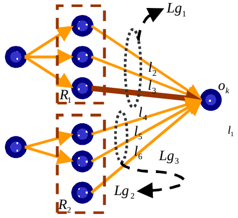
w_i : Worst case runnable queueing time of object o_i

$L_{s,t}$: End to end latency between object o_s and o_t

$$y_{h,k} = \begin{cases} 1, & \text{If activation of } o_k \text{ is event-driven by } o_h \\ 0, & \text{otherwise} \end{cases}$$

Feasibility Constraints 1

Jitter Inheritance Rule



$$y_{r,k} = y_{s,k}$$

All links in one group assume the same activation model

$$\sum_{Lg_h \in G(o_k)} y_{r,k} \le 1$$

Only one of the incoming link group can provide its activation signal

$$J_k \le \sum_{Lg_h \in G(o_k)} y_{r,k} \times M$$

$$0 \le J_k$$

If none of incoming groups carry activation signal, then release jitter of object k is 0

$$J_k \le r_r + (1 - y_{r,k}) \times M$$

$$r_r - (1 - y_{r,k}) \times M \le J_k$$

Release jitter inherited from object r which has largest wcrt from the activating group

$$r_h + (y_{h,k} - 1) \times M \le J_k$$

$$J_k \le r_h$$

$$J_k \le y_{h,k} \times M$$

Simplified version of link groups

Feasibility Constraints 2

WCRT Rule

$$r_h = w_h + J_h$$

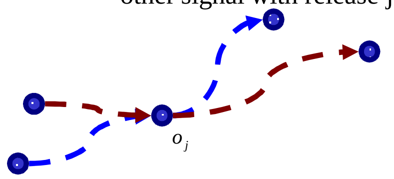
$$w_h = C_h + \sum_{k \in hp(h)} \left(\frac{w_h + J_k}{T_k} + \alpha\right) C_k$$

$$\sum_{P_r \in P} (\alpha \times L_{P_r}^{\uparrow} + (1-\alpha) \times L_{P_r}^{\downarrow} - L_{p_r})^2$$

Calculation of worst case response time

- A linear combination of linear upper and lower bounds is used as an estimation of runnable queuing time
- alpha is chosen to minimize the mean square fit function

$$z_{i,j} = \begin{cases} w_j & \text{if link } l_{i,j} \text{ carries activation signal} \\ w_j + J_j + T_j & \text{otherwis, } o_j \text{ may be activated by} \\ & \text{other signal with release jitter } J_j \end{cases}$$



Path end to end latency can not exceed deadline

Latency Rule

$$w_j \le z_{i,j}$$

$$z_{i,j} \le w_j + (1 - y_{i,j}) \times M$$

$$z_{i,j} \le w_j + J_j + T_j$$

$$w_j + J_j + T_j - y_{i,j} \times M \le z_{i,j}$$

$$L_{s,t} = \sum_{l_{u,v} \in P_{s,t}} z_{u,v}$$

$$L_{s,t} \le d_{s,t}$$

Possible Objective Function

$$Maximize \sum_{Lg_h \in G} y_{j,k}$$

Minimization of the number of event buffers in the system

$$Minimize \sum_{P_r \in P} L_{p_r}$$

Minimization of sum of end to end latencies

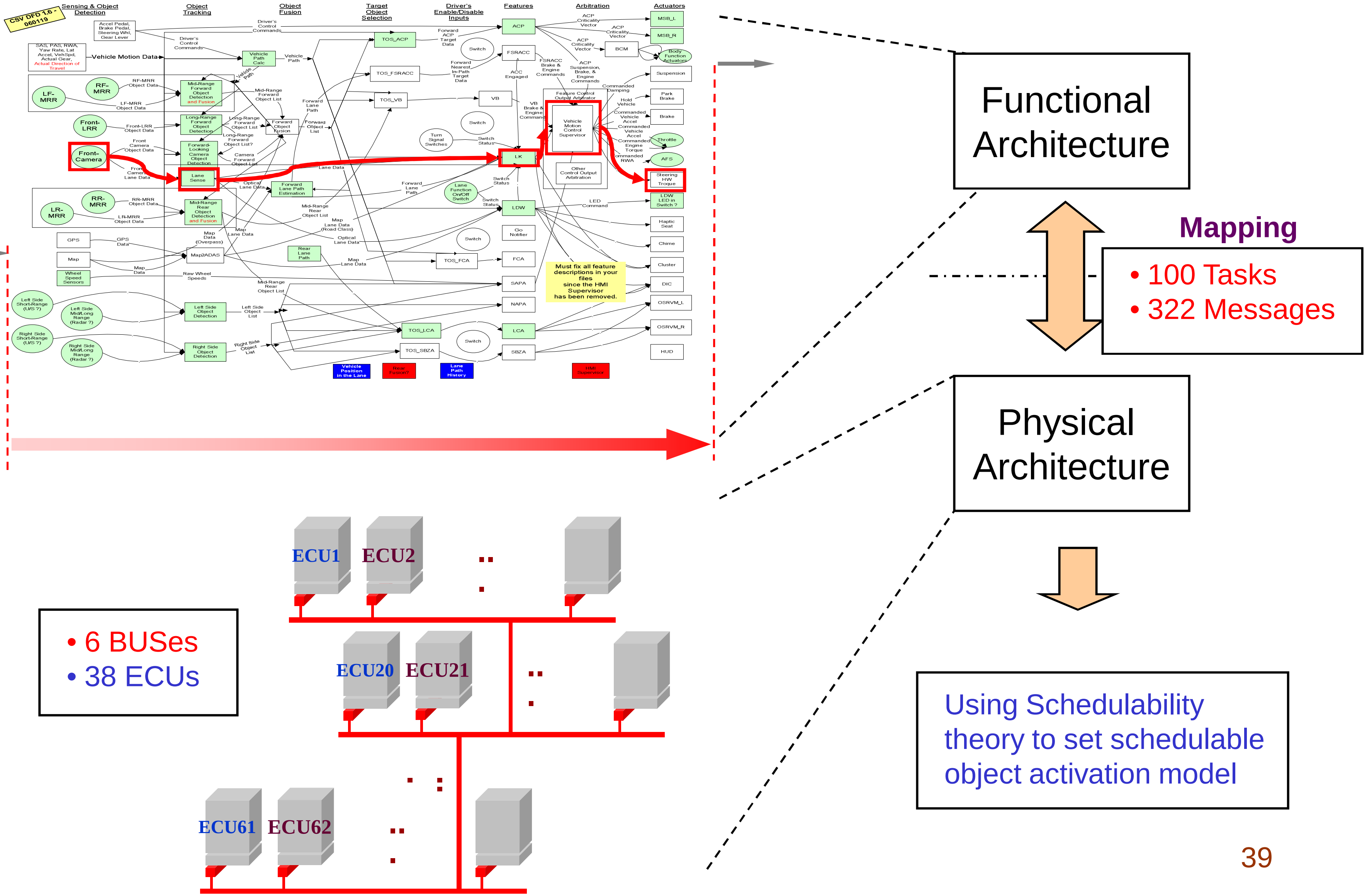
$$Minimize \sum_{P_r \in P} \gamma_{p_r} \times Max(L_{p_r} - d_{p_r}, 0)$$

Minimization of sum of weighted deadline violation

Outline

- Automotive architecture trends and challenges
- System-level design methodology and timing evaluation
- Activation models and end-to-end latencies
- Problem definition
  - Example
- MILP Optimization
- Case Study

Experimental vehicle case study



Functional Architecture

Mapping

- 100 Tasks
- 322 Messages

Physical Architecture

- 6 BUSes
- 38 ECUs

Using Schedulability theory to set schedulable object activation model

Case study results

Before Optimization
(all periodic)

- **Worst case = 577ms was found for paths with deadline 300ms**
- **Worst case = 255.5ms found for paths with deadline 200ms**
- **Worst case = 145.4ms found for paths with deadline 100ms**

Problem characterization

- **38 ECUs, 6 Buses**
- **Bus speed between 25 and 500 kb/s**
- **Bus utilization between 30% to 50%**
- **CPU utilization between 5% to 60%**
- **100 tasks, 322 messages**
- **Number of links in the functional dataflow is 507**
- **184 Paths analyzed between 10 pairs of functional nodes**

Optimization results

- **A feasible solution is found if using the largest lateness path metric**

after changing 24 groups

- **294.8 for paths with d=300**
- **158.1 for paths with d=200**
- **95.46 for paths with d=100**

(61.57 average slack)

- **the solution was improved with 5 extra branches**

(76.79 average slack)

α practically constant =0.465

with weighted sum of path latencies (evaluating all nodes) no solution found

Time to solve is

- **2.6 s for the exact analysis**
- **7 s for the linear approx**

(on a 1.4GHz PC)

Approach

- Mathematical programming
  - Modifying an object period affects multiple paths
  - Additional constraints due to legacy tasks and messages
- Geometric Programming: Poly-time optimization
  - Standard Form:

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_i(x) \leq 1 & i = 1, \ldots, m \\ & g_i(x) = 1 & i = 1, \ldots, p \end{array}$$

  - $x = (x_1, x_2, \ldots, x_n)$ are positive real-valued variables
  - *g* is a set of monomial functions

$$m(x) = c x_1^{a_1} x_2^{a_2} \ldots x_n^{a_n} \qquad c > 0, a_i \in \mathbb{R}$$

  - *f* is a set of posynomial functions
    - Sum of monomials

Geometric programming formulation

- Approximate the response time r_i with s_i
  - $0 \leq \alpha_i \leq 1$
  - If all $\alpha_i = 1$, $s_i \geq r_i$

$$\boxed{s_i = c_i + \sum_{j \in hp(i)} \left(\frac{s_i}{t_j} + \alpha_i \right) c_j \quad \forall o_i \in \mathcal{T}}$$

Minimize the sum of approx. response times

$$\text{min.} \quad \sum_{o_i \in \mathcal{O}} s_i$$

Meet end-to-end latency deadlines

$$\text{s.t.} \quad \ell_p \leq d_p \qquad \forall p \in \mathcal{P}$$

Transformed equations for approx. response times

$$\left\{ \begin{array}{ll} \frac{\sum_{j \in hp(i)} c_j \alpha_i + c_i}{s_i} + \sum_{j \in hp(i)} \frac{c_j}{t_j} \leq 1 & \forall o_i \in \mathcal{T} \\ \frac{\sum_{j \in hp(i)} c_j \alpha_i + b_i}{s_i'} + \sum_{j \in hp(i)} \frac{c_j}{t_j} \leq 1 & \forall o_i \in \mathcal{M} \\ s_i = s_i' + c_i & \forall o_i \in \mathcal{M} \end{array} \right.$$

Ensure schedulability

$$\frac{s_i}{t_i} \leq 1 \qquad \forall o_i \in \mathcal{O}$$

Meet utilization bounds

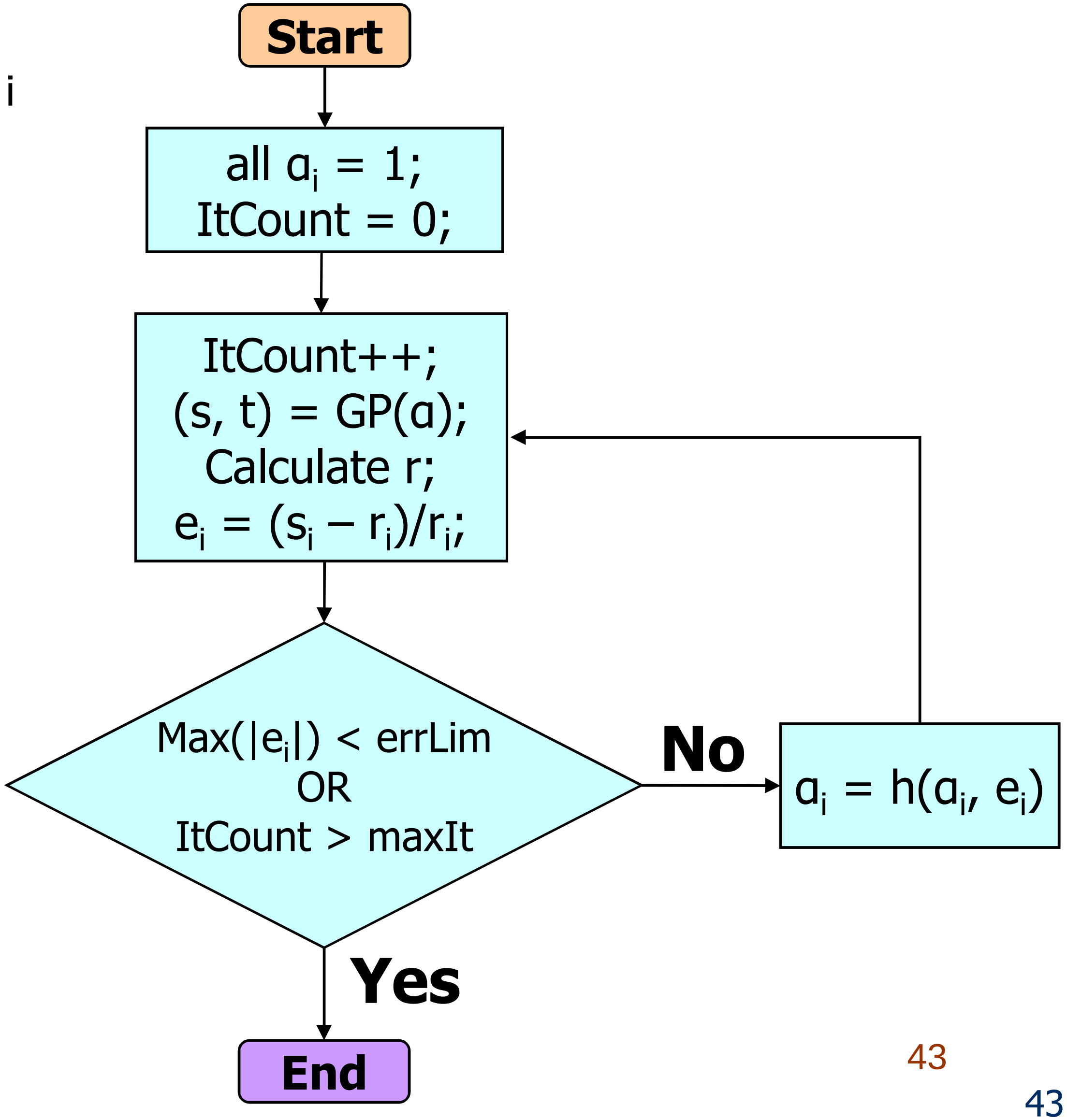
$$\sum_{o_i | R_{o_i} = j} \frac{c_i}{t_i} \leq u_j \qquad \forall R_j \in \mathcal{R}$$

Lower and upper bounds for periods

$$n_i \leq t_i \leq x_i \qquad \forall o_i \in \mathcal{O}$$

Iterative Procedure to Reduce Error

- Iteratively change α_i based on error
- Parameters
  - maxIt – max. # of iterations
  - errLim – max. permissible error



Start

all $\alpha_i = 1$;
ItCount = 0;

ItCount++;
(s, t) = GP(α);
Calculate r;
$e_i = (s_i - r_i)/r_i$;

Max($|e_i|$) < errLim
OR
ItCount > maxIt

No

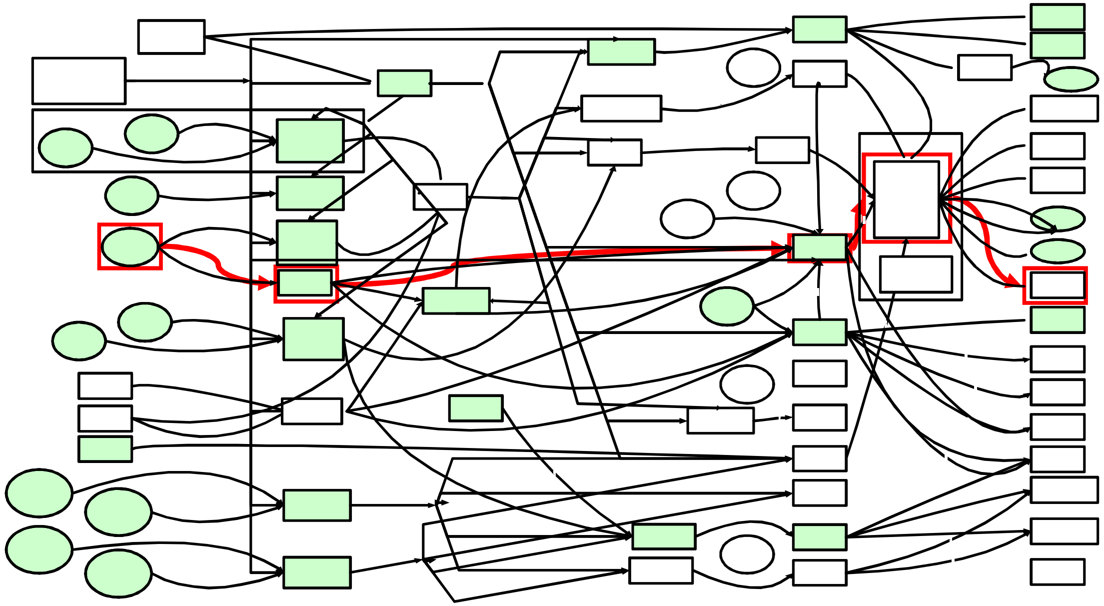
$\alpha_i = h(\alpha_i, e_i)$

Yes

End

Case Study: Advanced Safety Vehicle

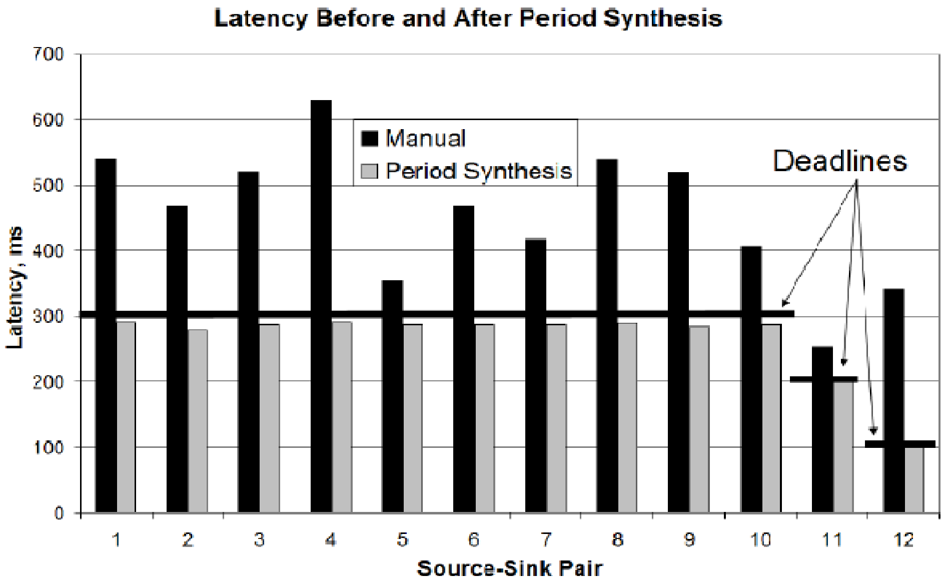
- From GM Research
- E.g. enhanced cruise control, lane departure warning, parallel parking assist



- Architecture
  - 38 ECUs
  - 4 buses
- Functionality
  - 92 tasks
  - 196 messages

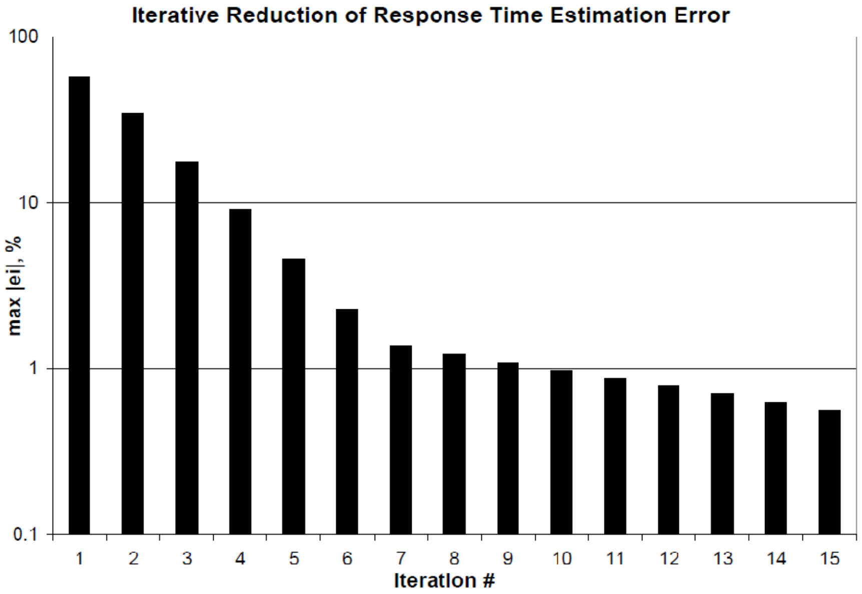
- End-to-end latency constraints
  - Over 12 source-sink task pairs
  - 222 total paths
  - Deadlines range from 100ms to 300ms

Experiments


Latency Before and After Period Synthesis

- GP optimization meets all deadlines in 1st iteration
- Solution time: 24s

- Maximum error reduced from 58% to 0.56% in 15 iterations
- Average error (not shown) reduced from 6.98% to 0.009%


Iterative Reduction of Response Time Estimation Error

Concluding remarks

- Quantitative analysis offers opportunities for architecture exploration and selection
- Domains of cost, dependability and time have been identified as prime candidates
  - not considering, for example, power
- Analysis techniques are at different levels of maturity
- Uncertainty challenge
  - Some required information is typically not available in the early development stages
  - Requirements extraction process is not mature
- Synthesis to be extended to other domains
  - leveraging MILP or GP formulations of the placement, priority assignment and period definition problems

Concluding remarks

- Worst case timing analysis can be applied to design optimization problems
- With respect to end-to-end latencies in distributed architectures there are multiple dimensions that can be explored
  - task allocation
  - period assignment
  - priority assignment
  - ...
- Also, most active safety functions are not truly hard real-time and worst case analysis may be pessimistic
  - end-to-end stochastic analysis
  - design optimizations based on stochastic analysis ?

Q&A

Thank you!

E-mail: marco@sssup.it