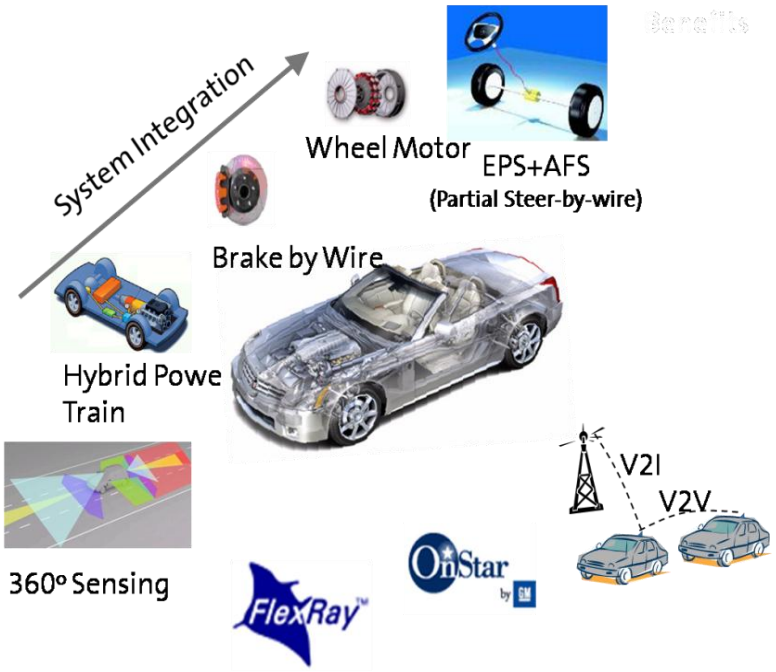
Early Time-Budgeting in Distributed Embedded Control Systems

Manoj G. Dixit,

Jointly with,

Prof. Pallab Dasgupta, IIT KGP and Dr. S. Ramesh, GM R&D

Background

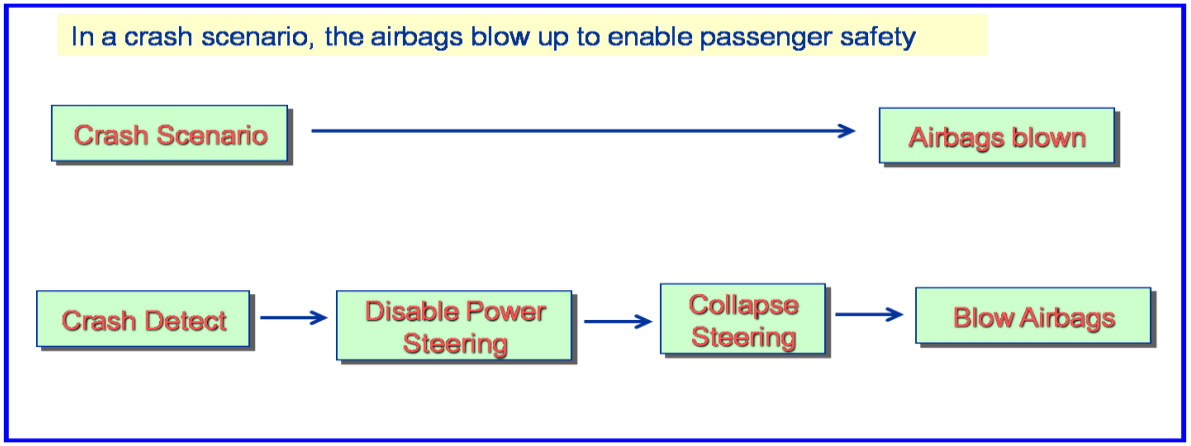


- Modern embedded control systems are component based and have large number of features
  - E.g. AUTOSAR based development
- Safety features have strict real-time end-to-end requirements
- Many components interact together to meet system level requirements
- System is distributed in nature

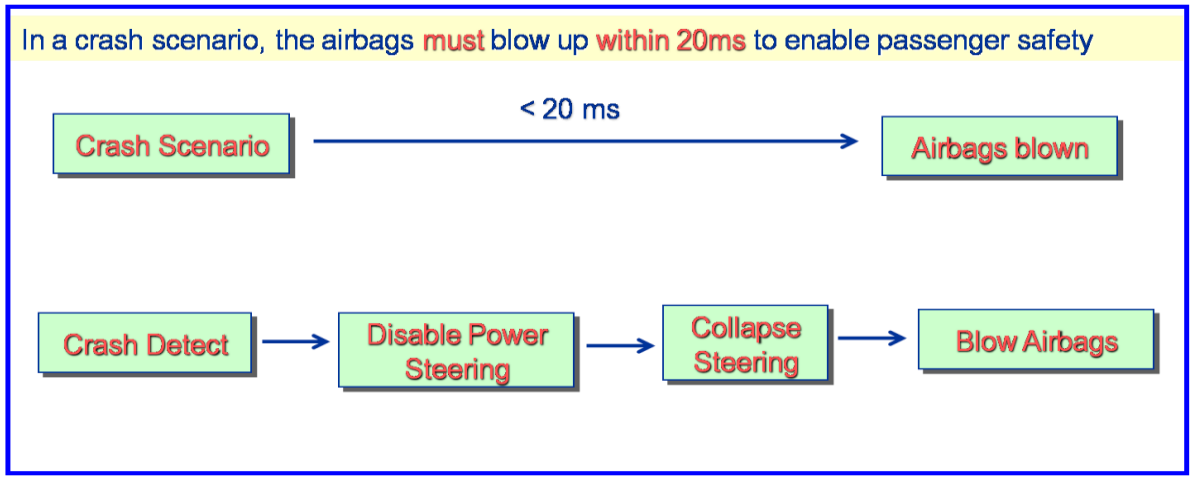
How to do a timing layout of entire system to meet end-to-end real-time requirements?

Illustrative Example

Functional Specification

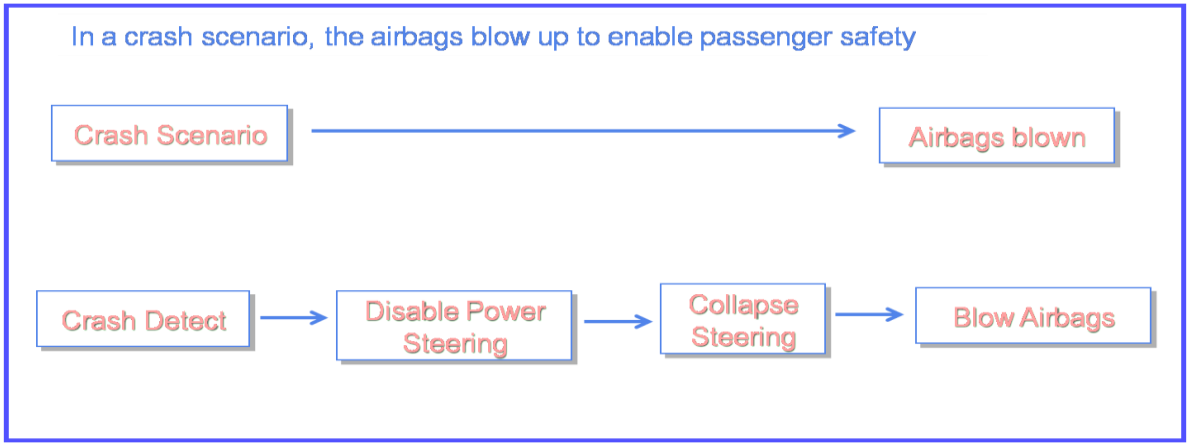


Safety Requirement



illustrative Example

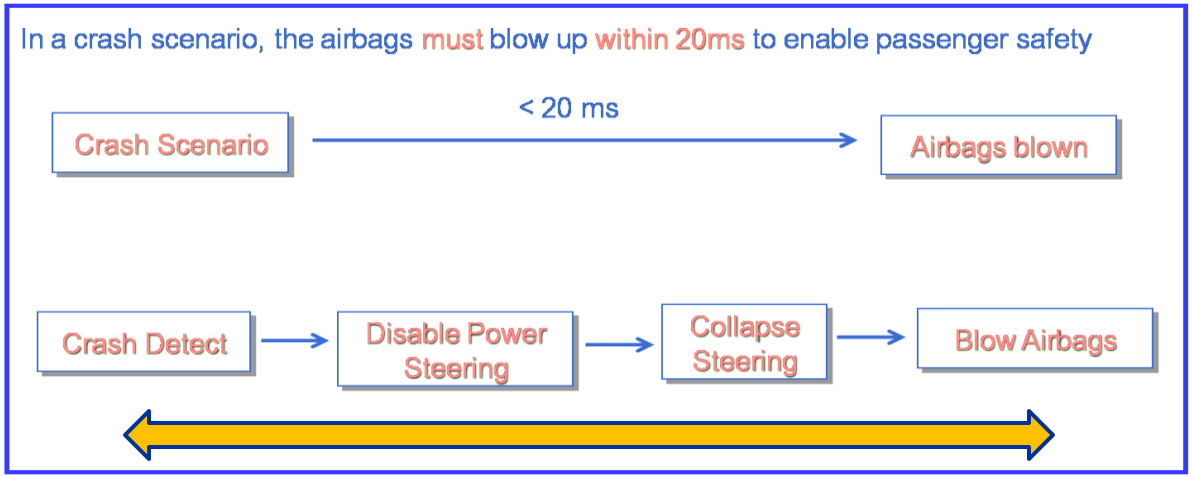
Functional Specification



Standards/Statistical data

Safety Requirement

Actual Implementation

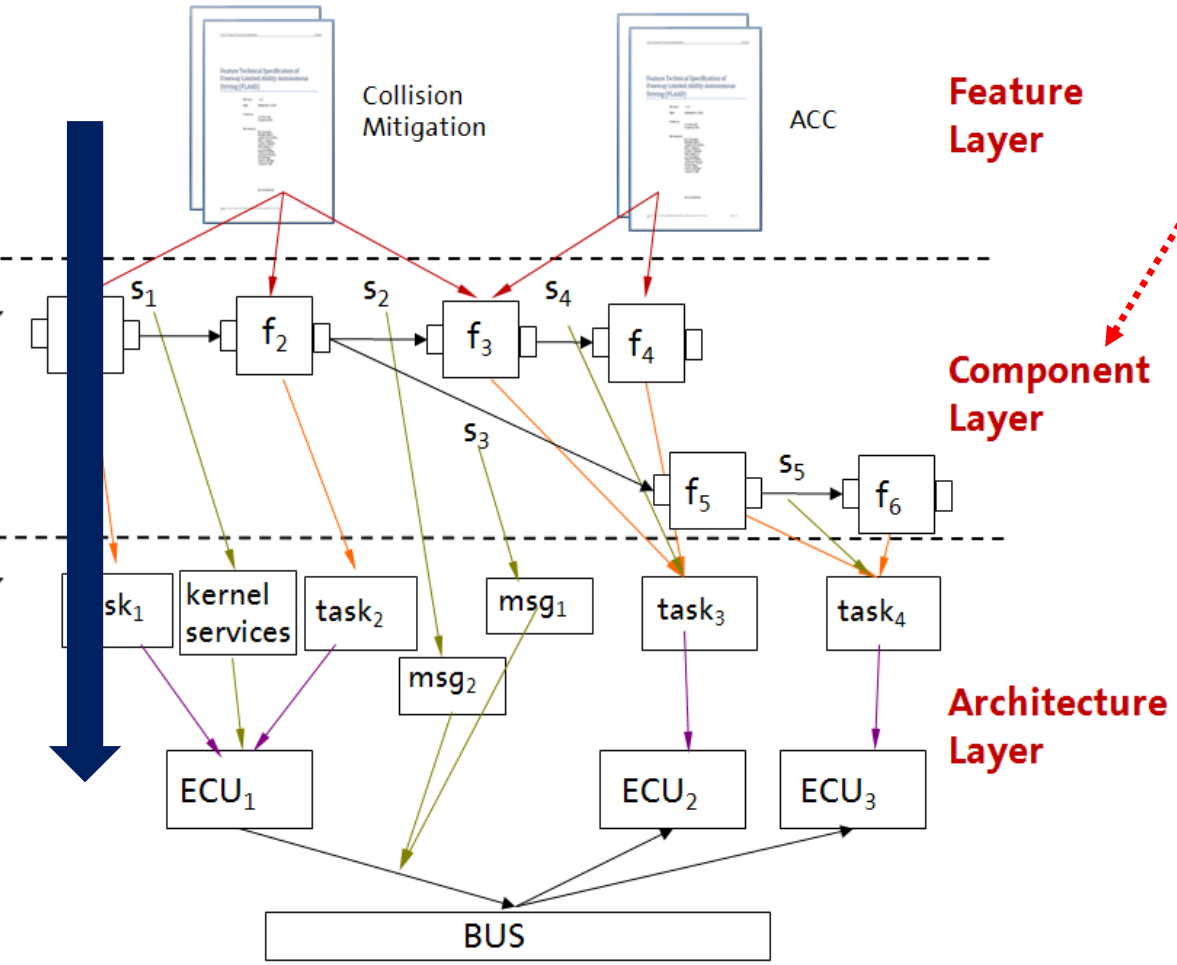


Time Budgeting

Emerging Challenges

- Increasing complex features
- Multiple functions in a single computational unit, e.g. AUTOSAR
- More component sharing promoted by the smaller component sizes
- Need for advance planning of resources for extensibility
- This is leading to...
  - *Increasing real-time interdependencies between components*

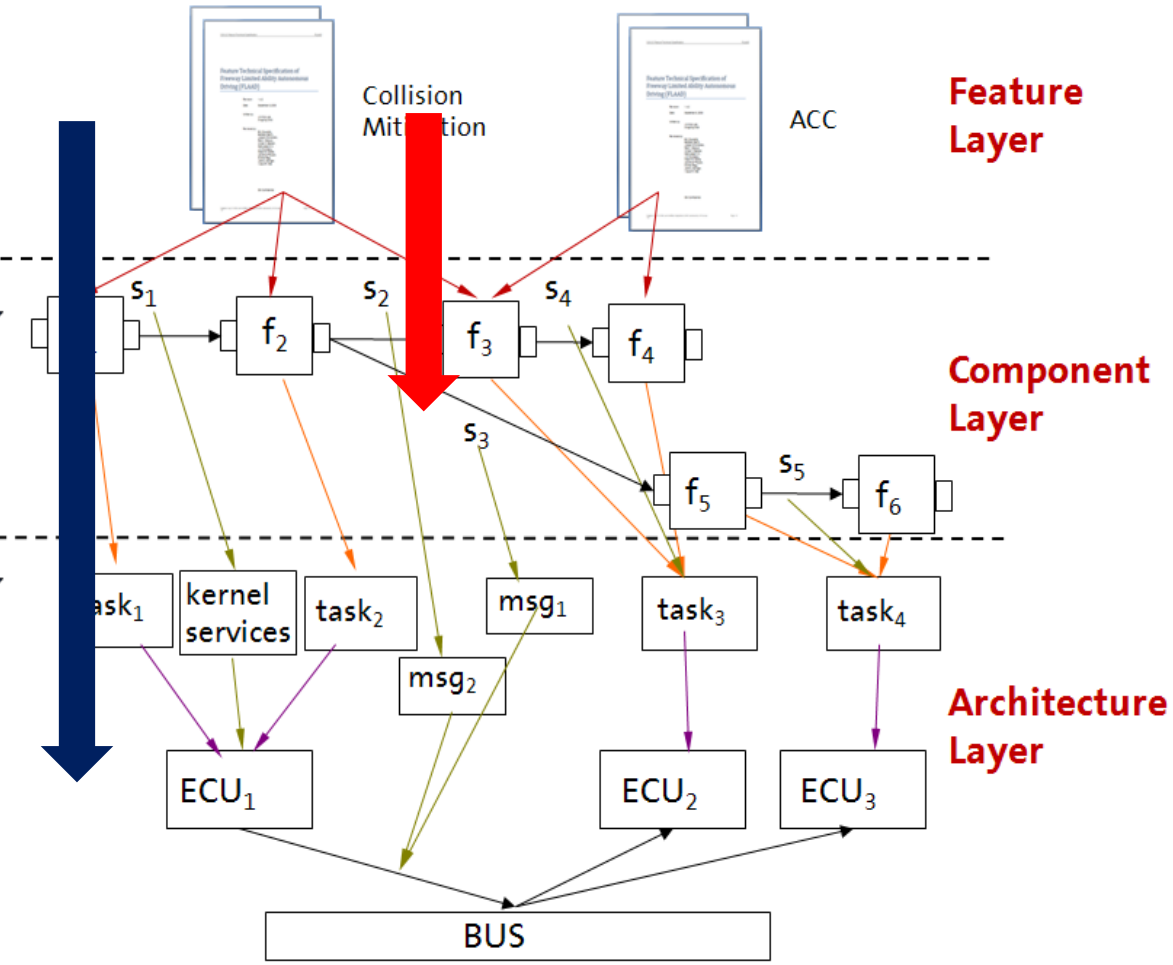
Prevalent Approaches



- Ad-hoc estimates about component response time
- Architecture exploration to do component-task mapping, component-ECU mapping etc.
- On failure, difficult to trace the culprit component

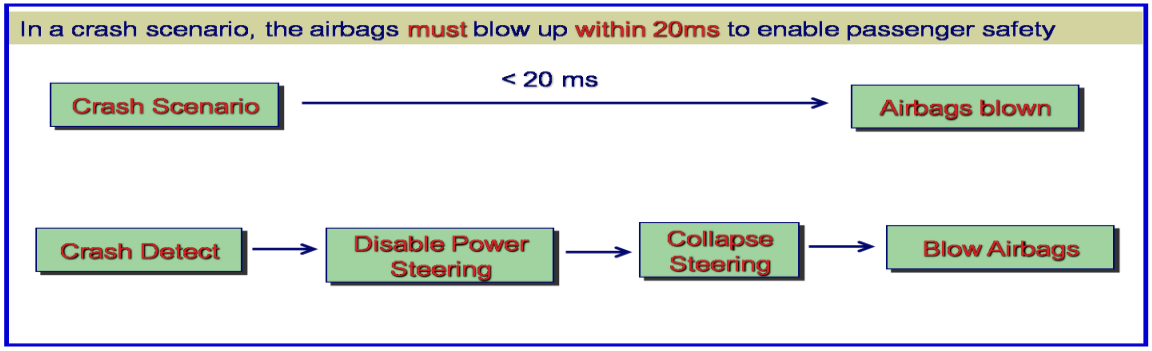
How to budget time for each component is not clear

Our Proposal



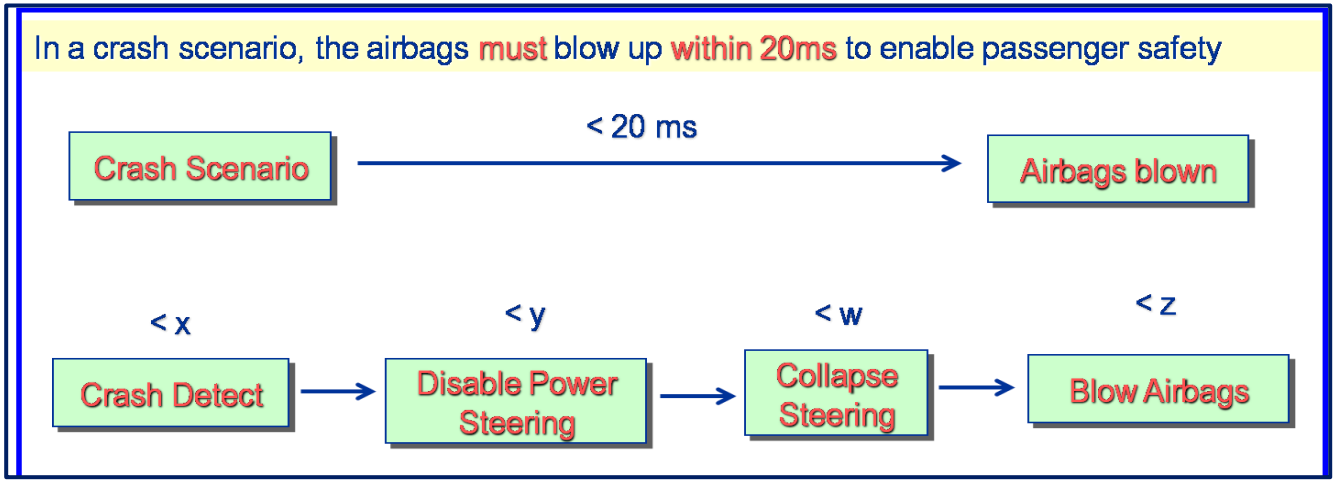
- Early time-budgeting for embedded control-systems
- *Component have parametric timing requirements*
- Use formal specification and analysis methodology to generate constraints on parameter valuations

Example



Real-time Specifications

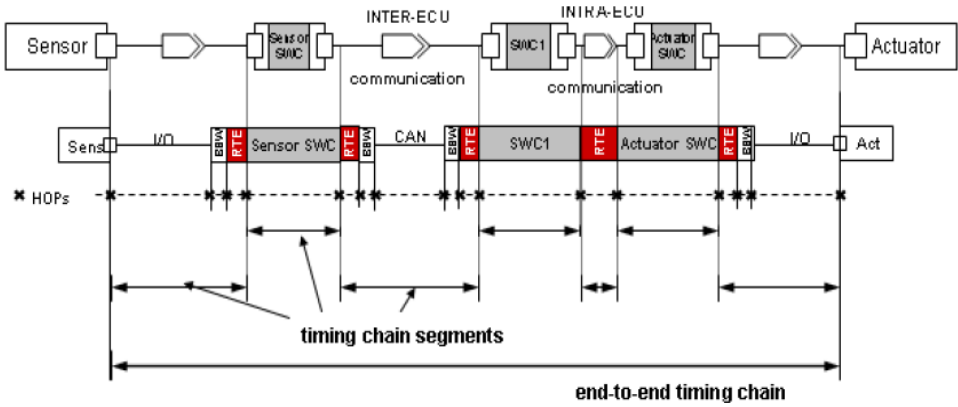
Parametric -time Specifications



Time-Budgeting: What values of x, y, w, z are good-enough?

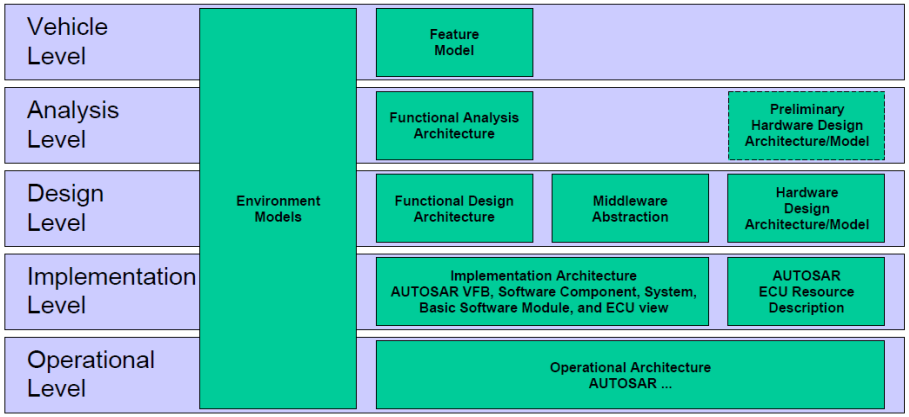
Industrial relevance

- Shift towards early specification of timing requirements
  - Large component integration, multiple suppliers
  - Important to know, how a specific choice of timing specification for one component affects the other



Source: Autosar timing spec from http://www.autosar.org

- AUTOSAR meta model allows specifying timing specifications at different levels of software hierarchy – components to network
  - EAST-ADL and TIMMO2 provides higher level of abstractions for specifying functional and product line requirements
  - Timing requirements are refined across different levels
    - Event models – periodic, sporadic etc.
    - Delay, synchronization constraints

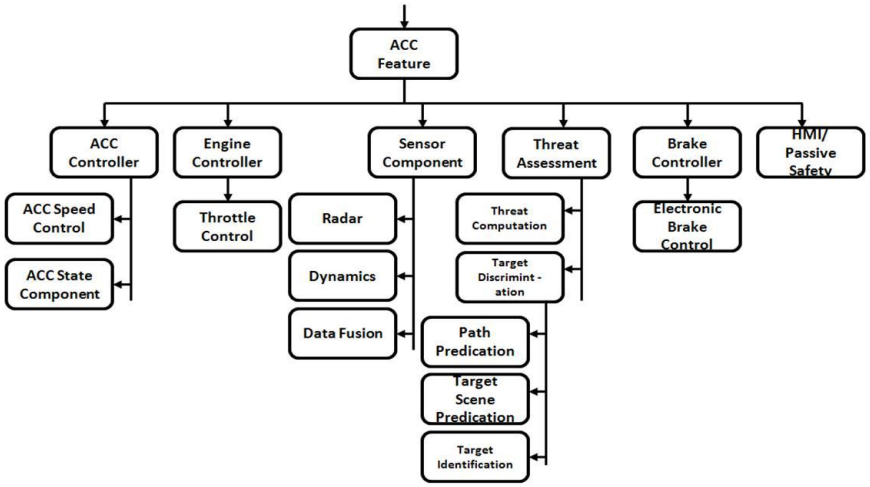


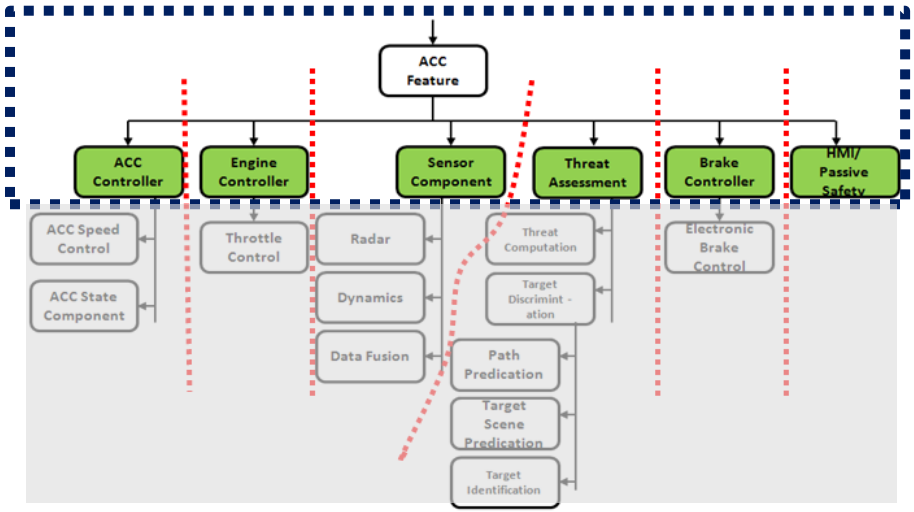
Source: TIMMO Methodology presentation by Stephan Kuntz, Continental Automotive GmbH, 2010

The Problem

- We are given a set of features and their real-time requirements
- We are given a set of components and their parametric-time requirements for implementing these features
- Propose Early stage Time-Budgeting Methodology
  - Find constraints over parameter values
  - Design space exploration to select suitable valuation
  - Scalable

Scalability and Usability Considerations -1

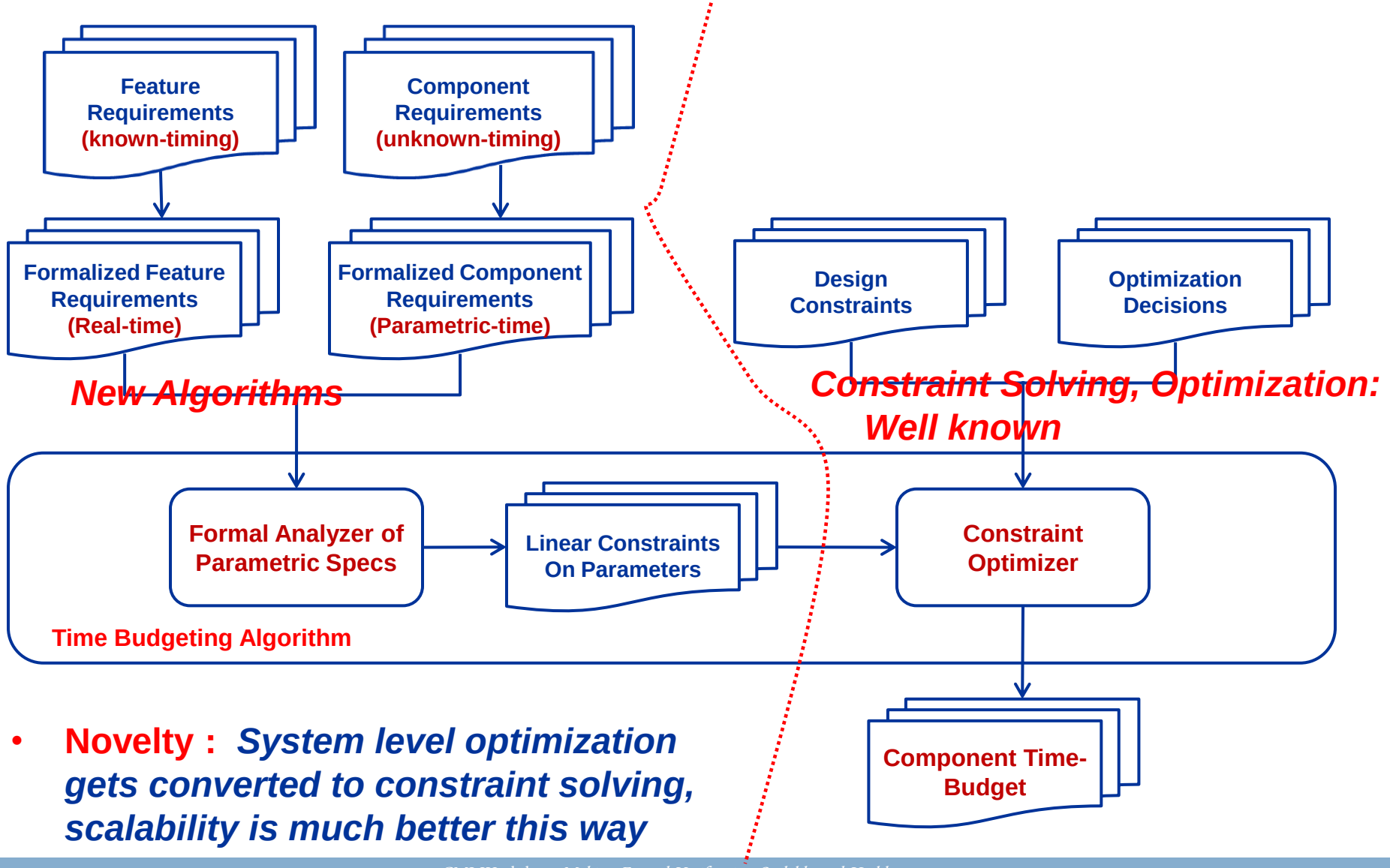




- In practice, component decompositions are hierarchical: DAG
  - Methodology aligned for hierarchical specifications
- Large decompositions:
  - Each feature component has 10s of requirements
  - Simultaneous budgeting does not seem to be scalable
  - Linear constraints are preferred
- Requirements become finer and more complex (or detailed) as we move down the hierarchy

- Handling large hierarchical decompositions:
  - Split component time-budgeting into smaller sub-problems and repeat
  - Compositional approach
  - DF-traversal with back-tracking takes care of component re-use case
- Specialized methods to analyze requirements patterns

Time-Budgeting Single Step

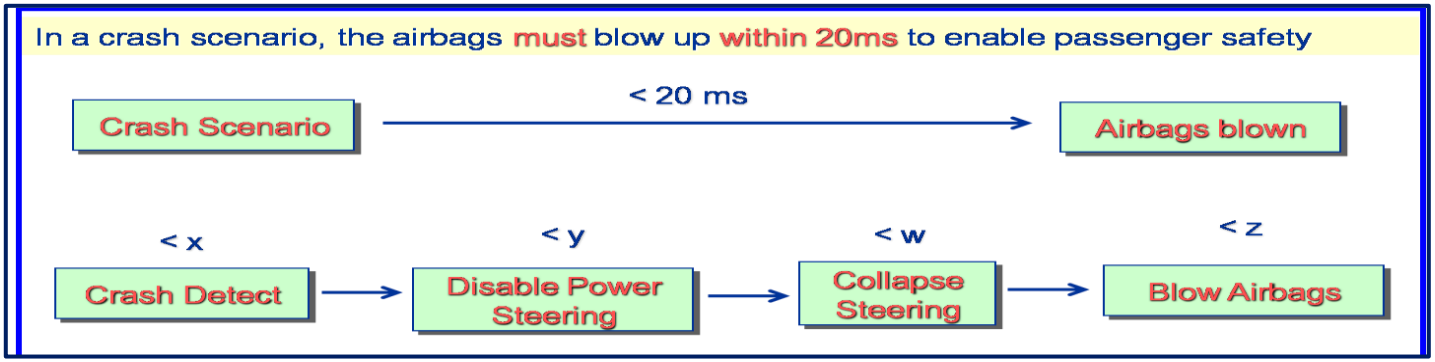


- **Novelty :** ***System level optimization gets converted to constraint solving, scalability is much better this way***

Formalization of Requirement Decomposition

- Requirements decomposition step is formalized as a collection of *requirement decomposition pairs*
- f: feature requirement and let $g_1, \ldots, g_k$ are component requirements identified for f
- $(f, \{g_1, \ldots, g_k\})$ is a requirement decomposition pair
- Verification check:
  - Informally: component requirements put-together should satisfy all feature requirements
  - We have the following reduction:

Theorem:
It is enough to analyze each pair separately, compute validity constraint. Any solution to conjoined constraint defines a suitable time-budget

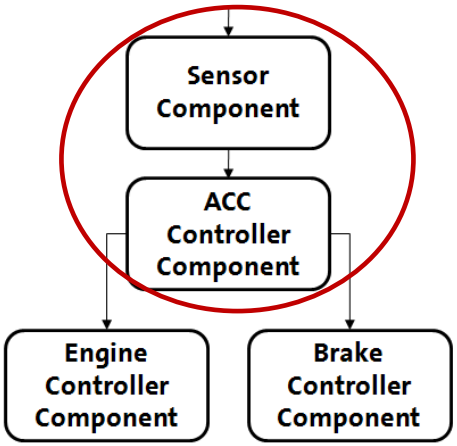


Formal Specification for Requirements

- Parametric Temporal Logic (PLTL)
  - Extends well known Linear Temporal Logic
  - Semantics is defined by using a parameter valuation

Feature/Component Name	PLTL Formula
ACC Feature	$\phi_1 : \Box(lead\_slow \Rightarrow \Diamond_{\leq 500}\ apply\_brake)$
Sensor Component	$\psi_1 : \Box(lead\_slow \Rightarrow \Diamond_{\leq x_1}\ lead\_kinematics\_info)$
ACC Controller Component	$\psi_2 : \Box(lead\_kinematics\_info \Rightarrow \Diamond_{\leq x_2} apply\_brake)$

$(\phi_1, \{\psi_1, \psi_2\})$ *is a Requirement Decomposition Pair*



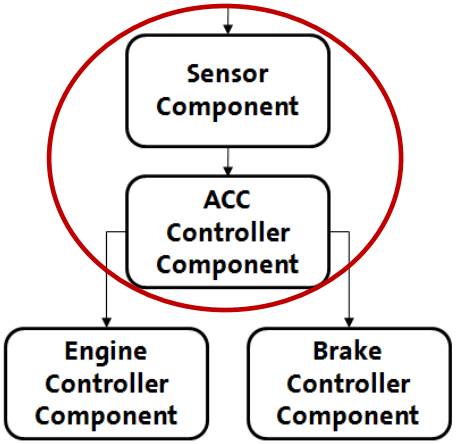
Validity of a Requirement Decomposition Pair

Feature/Component Name	PLTL Formula
ACC Feature	$\phi_1 : \Box(lead\_slow \Rightarrow \Diamond_{\leq 500}\ apply\_brake)$
Sensor Component	$\psi_1 : \Box(lead\_slow \Rightarrow \Diamond_{\leq x_1}\ lead\_kinematics\_info)$
ACC Controller Component	$\psi_2 : \Box(lead\_kinematics\_info \Rightarrow \Diamond_{\leq x_2} apply\_brake)$

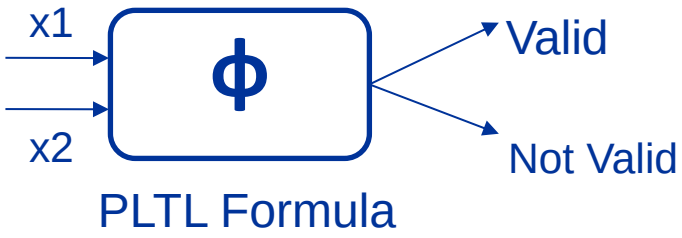
The requirement decomposition pair is *valid* if and only if

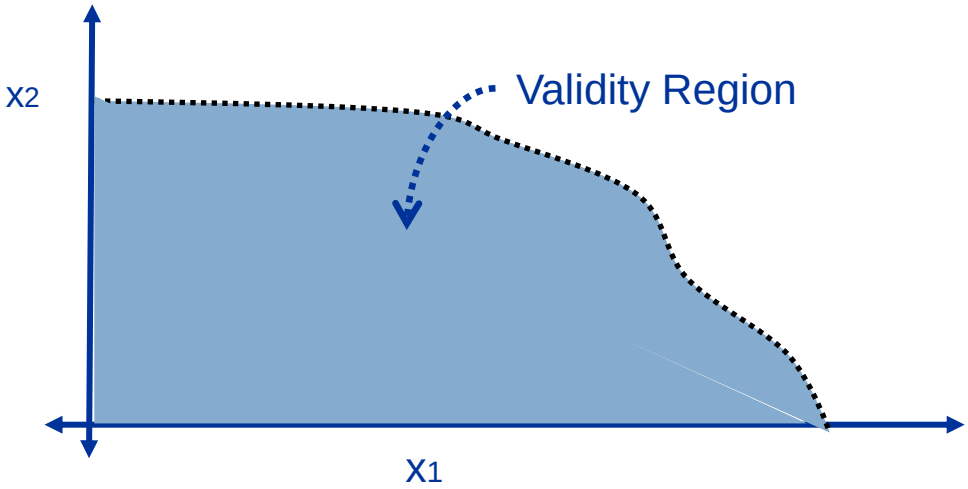
PLTL formula $\psi_1 \wedge \psi_2 \Rightarrow \phi_1$ is valid.

Due to the parameters, this reduces to constraint computation



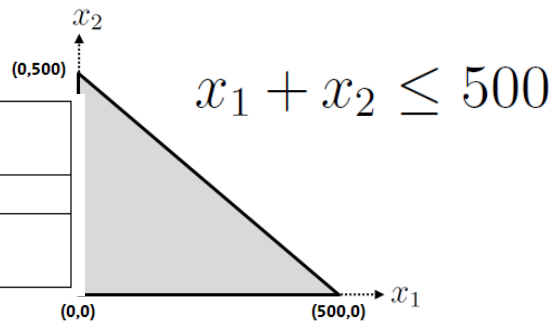
Abstractly…





Given a PLTL formula ϕ, we want to find the representation of the solution region in the form of a constraint

Feature/Component Name	PLTL Formula
ACC Feature	$\phi_1 : \Box(lead\_slow \Rightarrow \Diamond_{\leq 500}\ apply\_brake)$
Sensor Component	$\psi_1 : \Box(lead\_slow \Rightarrow \Diamond_{\leq x_1}\ lead\_kinematics\_info)$
ACC Controller Component	$\psi_2 : \Box(lead\_kinematics\_info \Rightarrow \Diamond_{\leq x_2} apply\_brake)$



Scalability and Usability Considerations - 2

- Scalable decision procedures for PLTL
  - Emptiness, universality conditions for a formula
  - Closed form representation of validity region possibly using linear constraints
  - Constraint computation involves dealing with large search space to compute boundary

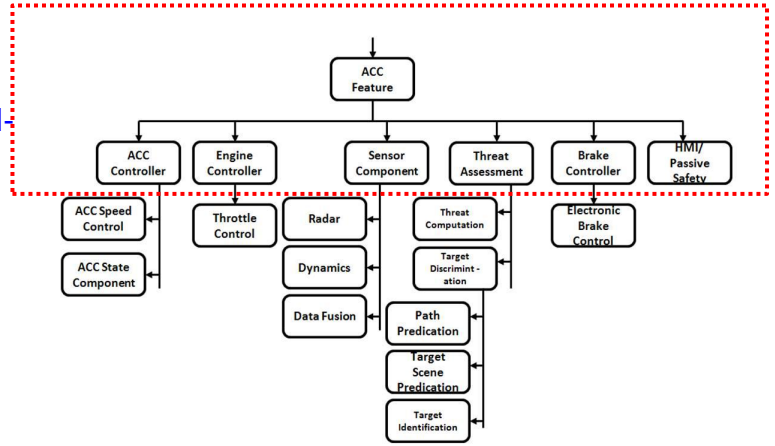
- Pattern specific scalable constraint computation techniques
  - Requirement decomposition pairs modeled using bounded-response pattern
  - Suitable for specifying end-to-end response

$$\Box(\phi \Rightarrow \Diamond_{\leq x}\psi)$$

Bounded-response pattern

- Developing modeling guidelines for enhancing the usability
  - Model requirements at top level of hierarchy using bounded-response pattern
  - At lower level have more complex pattern
  - Whenever scalability issues are encountered in constraint computation, perform bounded-response based decomposition and then refine

Model top requirements using bounded-response and then refine



Emptiness, Universality and Finiteness of Validity Region

+

Unsuitability of Linear Predicates for Representing Validity Region

Emptiness, Universality and Finiteness Problems

Formula Φ	Emptiness			Universality			Finiteness		
	F	S_Φ	V_Φ	F	S_Φ	V_Φ	F	S_Φ	V_Φ
PLTL$_\Box$	$\{\Phi(\alpha_0)\}$	$\Leftrightarrow$, sat	$\Leftrightarrow$, val	$\{\widetilde{\Phi}\}$	$\Rightarrow$, sat	$\Leftrightarrow$, val	$\{\widetilde{\Phi(0_{Y\setminus\{y\}})} \mid y \in Y\}$	$\Rightarrow$, any member sat	$\Leftrightarrow$, any member val
PLTL$_\Diamond$	$\{\widetilde{\Phi}\}$	$\Leftrightarrow$, sat	$\Rightarrow$, val	$\{\Phi(\alpha_0)\}$	$\Leftrightarrow$, sat	$\Leftrightarrow$, val	$\{\widetilde{\Phi}\}$	$\Leftrightarrow$, sat	$\Rightarrow$, val
PLTL	$\{\widetilde{\Phi(0_Y)}\}$	$\Leftrightarrow$, sat	$\Rightarrow$, val	$\{\widetilde{\Phi(0_X)}\}$	$\Rightarrow$, sat	$\Leftrightarrow$, val	$\{\widetilde{\Phi(0_Y)}\}$	$\Leftrightarrow$, sat	$\Rightarrow$, val

α_0:	Parameter valuation assigning 0 to all parameters.
0_X:	Partial parameter valuation assigning 0 to all members of X.
0_Y:	Partial parameter valuation assigning 0 to all members Y.
$0_{Y\setminus\{y\}}$:	Partial parameter valuation assigning 0 to all members of $Y \setminus \{y\}$.
$\Leftrightarrow$:	Solution provides both necessary and sufficient condition.
$\Rightarrow$:	Solution provides only sufficient condition.
sat:	Satisfiability check required of a member formula.
val:	Validity check required of a member formula.

- Defined *parameter abstraction operation* for PLTL

$$\Phi = \Box_{\leq y}\, p_1 \wedge \Box_{\leq 10}\, p_2 \qquad \widetilde{\Phi} = \Box\, p_1 \wedge \Box_{\leq 10}\, p_2$$

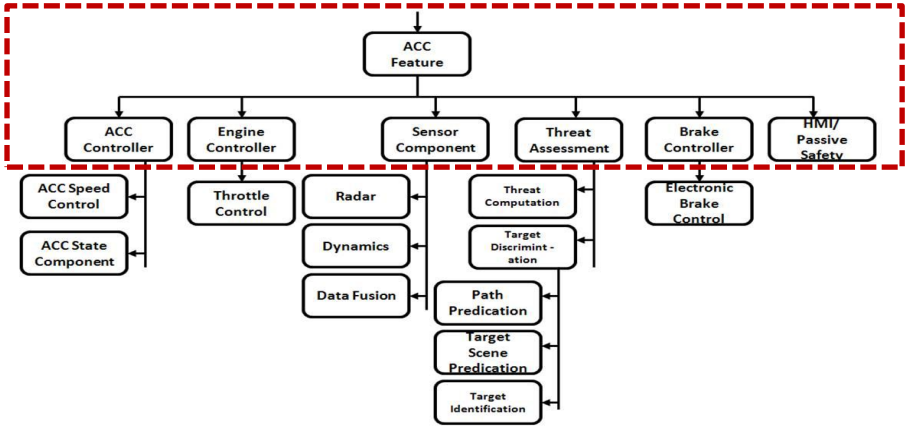
- Improved Complexity
  - Our complexity: $O(c_\Phi 2^{n_\Phi})$. Earlier: $O(c_\Phi^{k_\Phi+1} k_\Phi^{k_\Phi} 2^{n_\Phi(k_\Phi+1)})$

Unsuitability of Linear predicates

- Negative result
- Earlier known for a wider class of PLTL formulae
- We have further restricted this to a subclass of PLTL
- However.. all is not lost... many nice properties still fall in decidable fragment

Bounded-Response Constraint Extraction Method

A Scalable Method for a widely used Requirement Pattern



$$\Box(\phi \Rightarrow \Diamond_{\leq x}\psi)$$

Bounded-response formula in PLTL

- At higher levels most of the requirements are based on a specific pattern... *bounded-response*
- ϕ : Boolean formula
- ψ : Boolean formula
- x : Parameter or constant
- We consider validity checks of requirement decomposition pairs using this pattern

Constraint Extraction Method

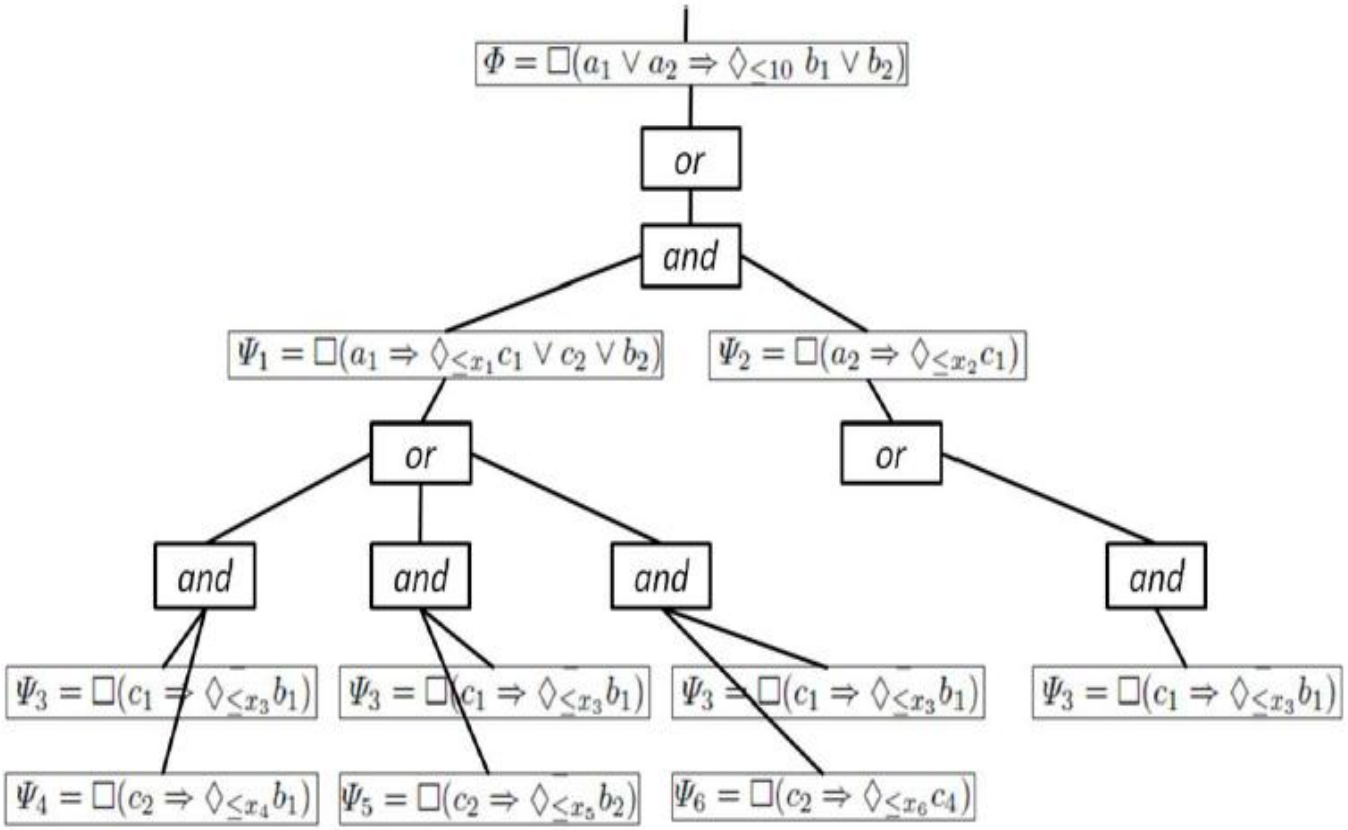
- Reasoning over temporal formulae reduced to Boolean reasoning... hence scalable
- And-or tree constructed from formulae
  - We have defined a notion of an *irreducible cover* for Boolean formulae
- Assign path constraints
- Final constraint: conjunction/disjunction of path constraints

Let γ and $\gamma_1, \cdots, \gamma_n$ be Boolean formulae.

An irreducible cover of γ is a minimal subset H of $\{\gamma_1, \cdots, \gamma_n\}$ such that $\gamma \Rightarrow \bigvee_{\gamma' \in H} \gamma'$ is valid.

Example: Bounded-Response Tree

$\Phi = \Box(a_1 \vee a_2 \Rightarrow \Diamond_{\leq 10}\ b_1 \vee b_2)$	$\Psi_1 = \Box(a_1 \Rightarrow \Diamond_{\leq x_1} c_1 \vee c_2 \vee b_2)$; $\Psi_2 = \Box(a_2 \Rightarrow \Diamond_{\leq x_2} c_1)$ $\Psi_3 = \Box(c_1 \Rightarrow \Diamond_{\leq x_3} b_1)$; $\Psi_4 = \Box(c_2 \Rightarrow \Diamond_{\leq x_4} b_1)$ $\Psi_5 = \Box(c_2 \Rightarrow \Diamond_{\leq x_5} b_2)$; $\Psi_6 = \Box(c_2 \Rightarrow \Diamond_{\leq x_6} c_4)$ $\Psi_7 = \Box(a_1 \wedge a_2 \Rightarrow \Diamond_{\leq x_7} c_3)$



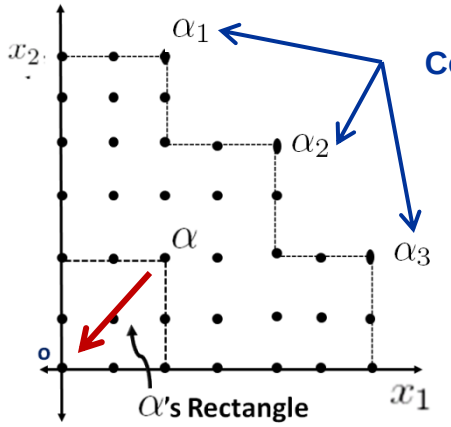
$$\{[(x_1+x_3 \leq 10) \wedge (x_1+x4 \leq 10)] \vee [(x_1+x_3 \leq 10) \wedge (x_1+x_5 \leq 10)] \vee [(x_1+x_3 \leq 10) \wedge \texttt{false}]\} \wedge (x_2+x_3 \leq 10)$$

Corner Point Constraint Extraction Method

A General Constraint Extraction Method for PLTL

$$\Box_{\leq m}\phi \Rightarrow \Box_{\leq (m-1)}\phi$$
$$\Diamond_{\leq m}\phi \Rightarrow \Diamond_{\leq (m+1)}\phi$$

Monotonicity of PLTL operators



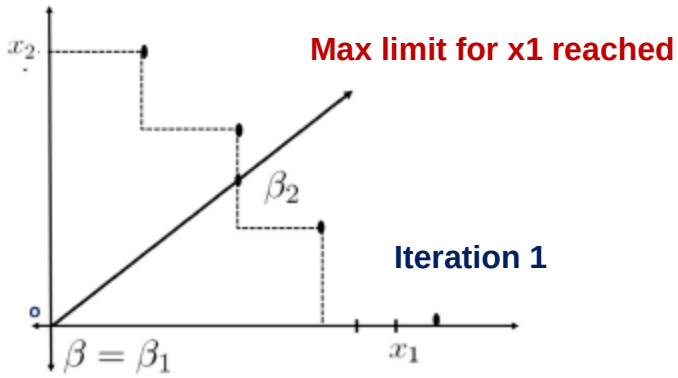
Downward Closure Property

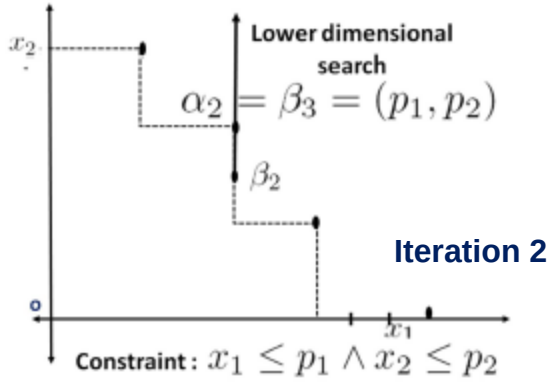
$$\bigvee_{i=1}^{3} x_1 \leq \alpha_i(x_1) \wedge x_2 \leq \alpha_i(x_2)$$

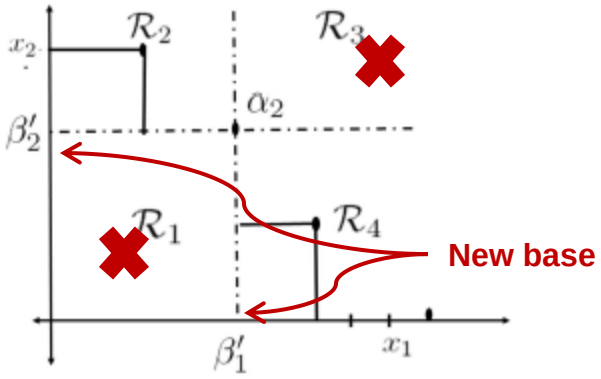
- Suitable for complex temporal properties
- We focus on PLTL fragments and their geometric properties
- PLTL Global fragment is downward-closed
- Downward Closed Region have **finite number** of *corner-points*
  - Include Point-at-infinity
  - Constraint definition using Corner-Points

Algorithm Overview

- Prune and Search Approach
  - Find a corner point
  - Partition the further search
- Search Step
  - Obtain a farthest useful valuation along diagonal starting from a base
  - Decide sub-set of parameters for which max limit is reached
  - Fix them and re-iterate till all parameters are over
- Prune Step
  - Partition and identify the region(s) where no corner-point can lie
  - Adjust new base valuation so that those regions get ignored from later search
  - Repeat Search step recursively for them

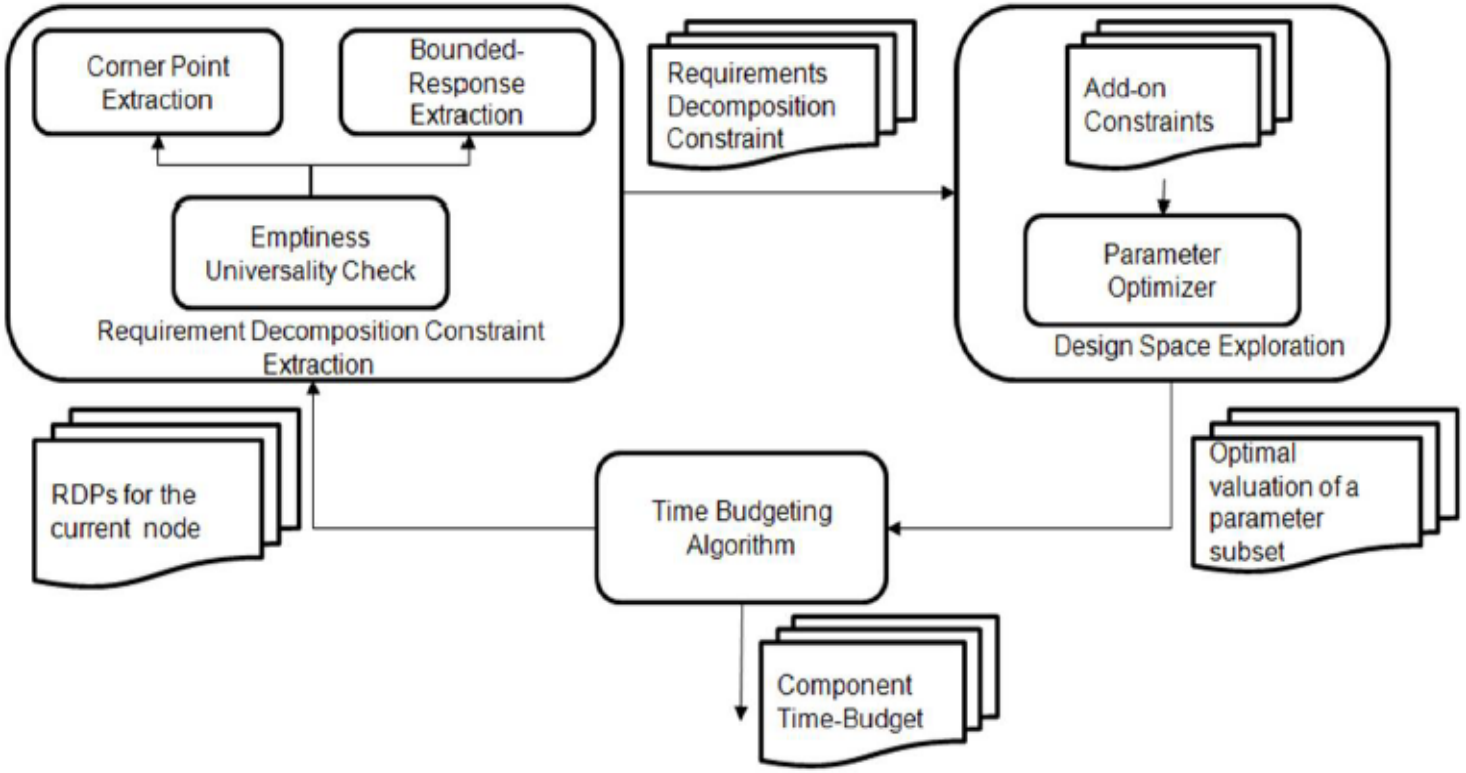






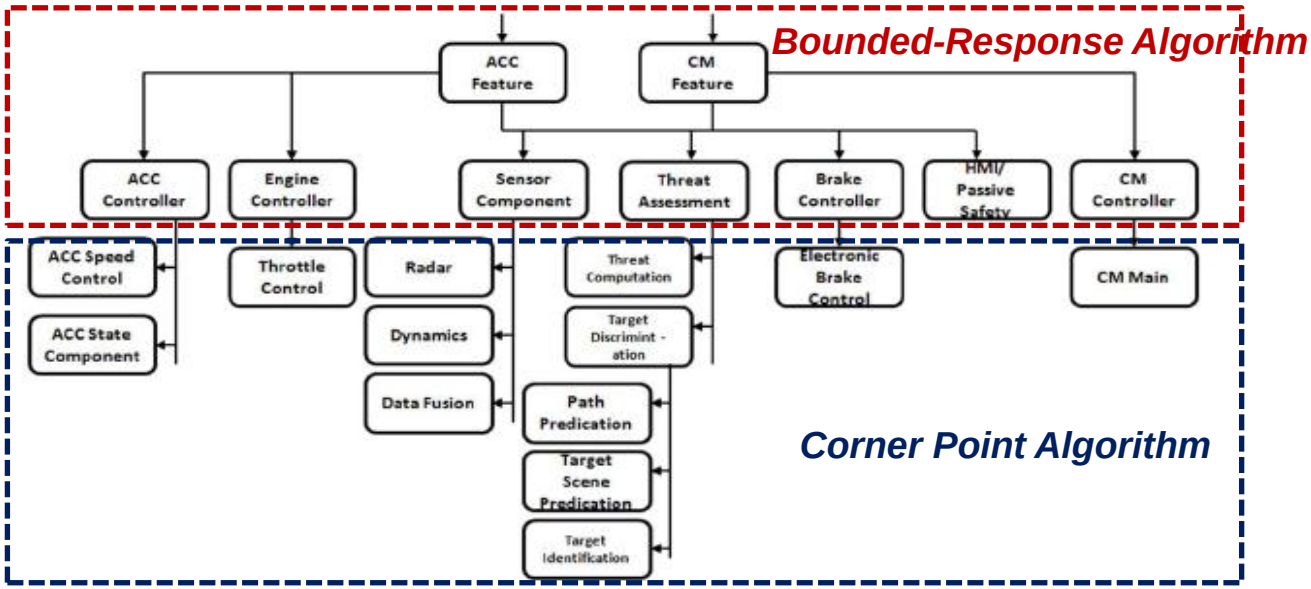
Demonstration of the Methodology

Integrated Time-Budgeting Methodology



- NuSMV for LTL checks
- Yices for constraint solving
- Eclipse, Java

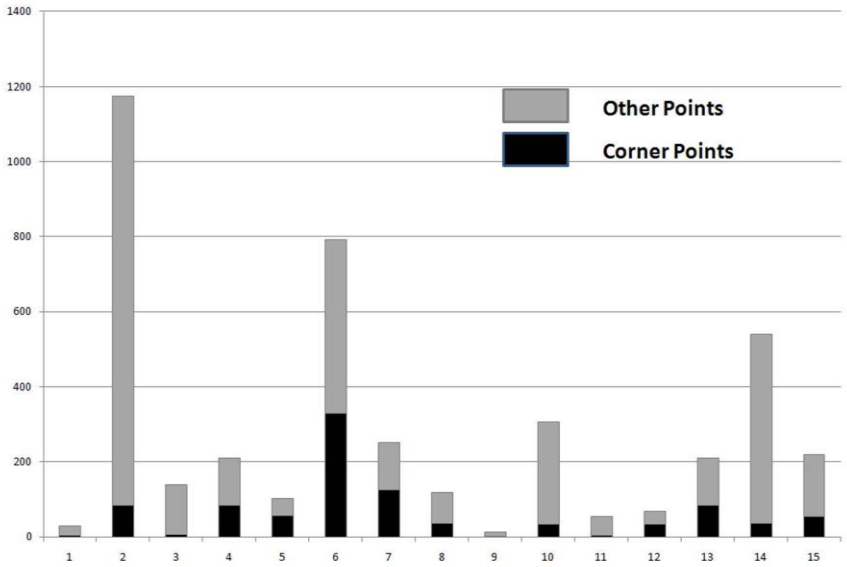
Case Studies



- Adaptive Cruise Control, Collision Mitigation
- 120+ feature and component properties
- 100+ add-on constraints in the design space exploration
- Budgeting for 3 feature combinations: ACC only, CM only and ACC-CM

Some Results

Component	Parameter	Only ACC	Only CM	ACC-CM
Brake Controller Component	w_1	200	160	160
	w_2	70	-	70
	w_3	100	-	100
	w_4	100	-	100
	w_5	10	-	10
Electronic Brake Control	w_6	150	100	100
	w_{11}	35	20	20
	w_{12}	100	80	80
	w_{13}	30	20	20
	w_{14}	10	40	20
	w_{15}	100	100	100
	w_{16}	60	-	60
	w_{17}	60	-	60



Decomposition Type	Num Components	RDP Nos	Num. SAT Checks (Average)	Constraint Computation Time (Average) (sec)
Single Level	12	15	314 (LTL)	1280
Multi Level	22	11 (Bounded-Resp), 20 (General)	83 (Boolean SAT), 57 (LTL)	2.36 (Boolean SAT), 19.25 (LTL)

Many Challenges still remain …

- Validity checking of parameter-free PLTL formulae
  - Presence of large constants lead to scalability of model checkers
- More scalable decidability algorithms for PLTL
- Seamless integration with architecture exploration phase to align with existing development flow

Summary

- A Hierarchical Time-Budgeting Methodology
  - Integrates all of the below techniques
- Emptiness, Universality and Finiteness Problems for PLTL
  - Non-triviality of the solution region
- Bounded-Response Constraint Extraction Method
  - A specially tuned method for a widely used requirements pattern
- Corner Point Constraint Extraction Method
  - Complex temporal relationships
- Case Studies
  - Tool framework and demonstration on automotive features ACC and CM

Publications

- Manoj G. Dixit, Pallab Dasgupta and S. Ramesh. *Taming the Component Timing: A CBD Methodology for Real-time Embedded Systems*. DATE, 1649-1652, 2010
- Manoj G. Dixit, S. Ramesh and Pallab Dasgupta. *Some Results on Parametric Temporal Logic*. Information Processing Letters, 994-998, 2011
- Manoj G. Dixit, S. Ramesh and Pallab Dasgupta. *Parametric Analysis of Real-time response guarantees on interacting software components*, World Intellectual Property Organization, WO/2009/129089, International Patent Application No: PCT/US2009/039837
- Manoj G. Dixit, S. Ramesh and Pallab Dasgupta. *Time-budgeting: A component based Methodology for Real-time Embedded Systems*. Accepted, Formal Aspects of Computing, Springer
- Manoj G. Dixit, S. Ramesh and Pallab Dasgupta, *Early Time-budgeting for Component based Embedded Control Systems*, Under Review, ESWEEK Workshop

References

R. Alur, K. Etessami, S. La Torre, and D. Peled. Parametric Temporal Logic for "Model Measuring". *ACM Transactions on Computational Logic*, 2(3):388–407, 2001.

C. Bartolini, G. Lipari, and M. Di Natale. From Functional Blocks to the Synthesis of the Architectural Model in Embedded Real-time Applications. In *IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 458–467, 2005.

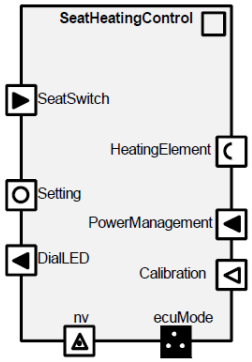
M. Di Natale and A. L. Sangiovanni-Vincentelli. Moving From Federated to Integrated Architectures in Automotive: The Role of Standards, Methods and Tools. *Proceedings of the IEEE*, 98(4):603–620, 2010.

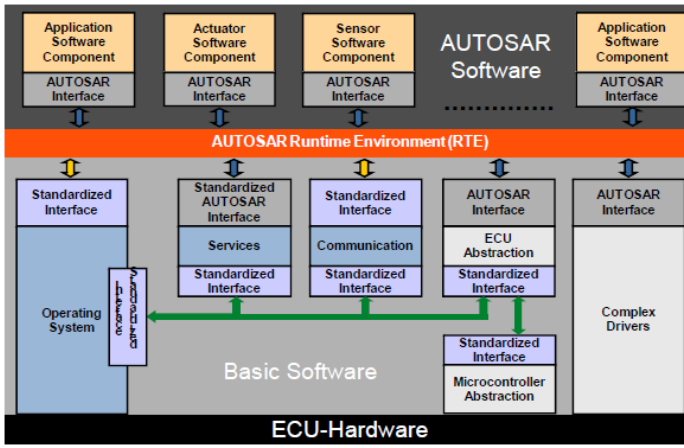
H. Blom, R. Johansson, and H. Lönn. Annotation with Timing Constraints in the Context of EAST-ADL2 and AUTOSAR - the Timing Augmented Description Language. In *Workshop on the Definition, evaluation, and exploitation of modelling and computing standards for Real-Time Embedded Systems*, 2009.

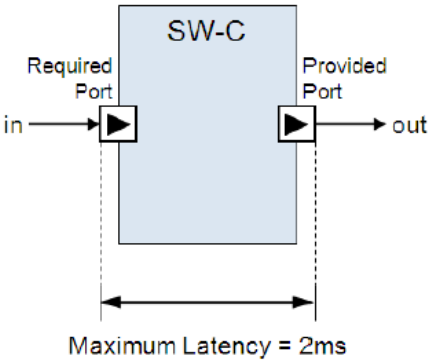
Thank You

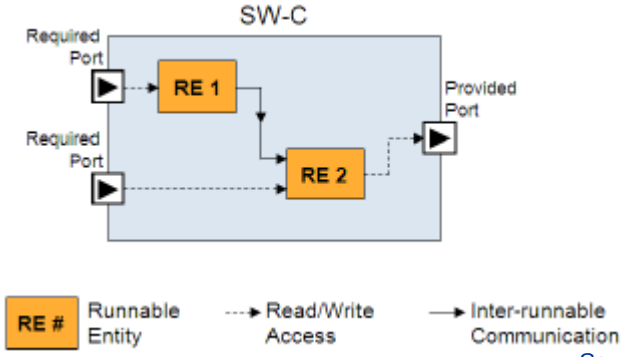
The relevance

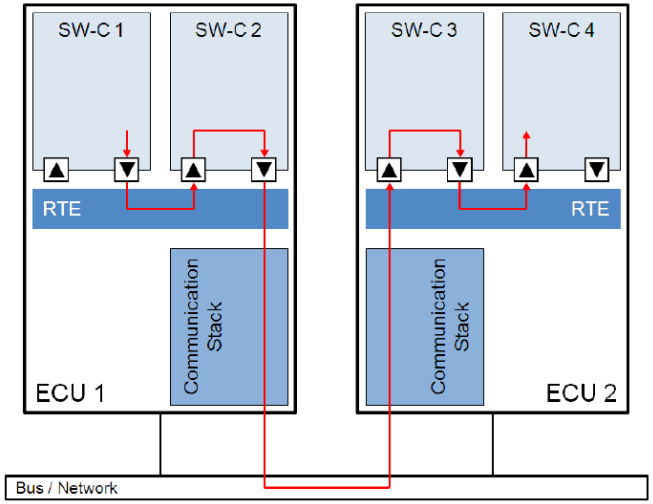
- AUTOSAR defines a meta-model for specifying component based distributed systems in automotive domain
- Specifications for components, middleware and higher level properties











Source: Autosar timing and VFBspecs from http://www.autosar.org