ITEA 2

INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

TIMMO

ITEA 2 – 06005: TIMMO
Timing Model

The TIMMO Methodology

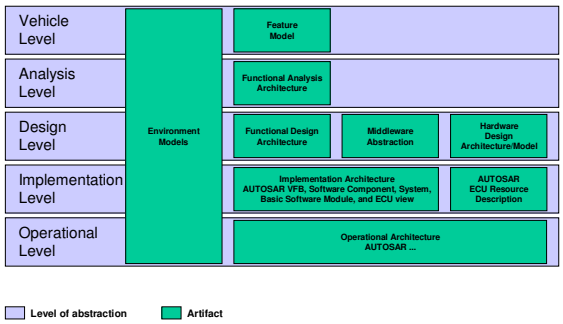
Guest lecture at Chalmers
February 8th, 2010

Daniel Karlsson, Volvo Technology AB

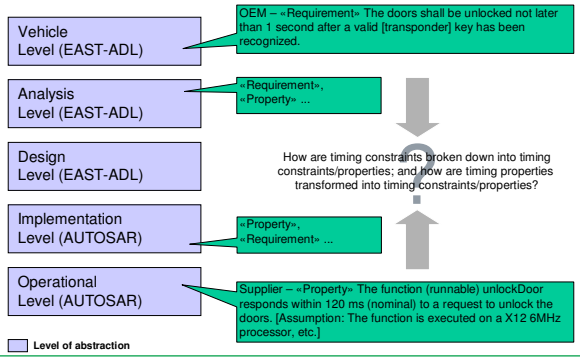
Objectives
TIMMO

- Solving the problem of describing the timing requirements imposed on and temporal behavior of a distributed real-time embedded software-intensive system
- Define a language to specify
  - timing requirements and constraints
  - timing properties
- Provide the capability to analyze and assess timing, a.k.a. temporal behavior, of a system beginning at early stages of the development process
- Define a methodology that enables one to apply the language in different scenarios
- Alignment with
  - **Aut**omotive **O**pen **S**ystem **Ar**chitecture - **AUTOSAR**
  - **EAST A**rchitecture **D**escription **L**anguage - **EAST-ADL**

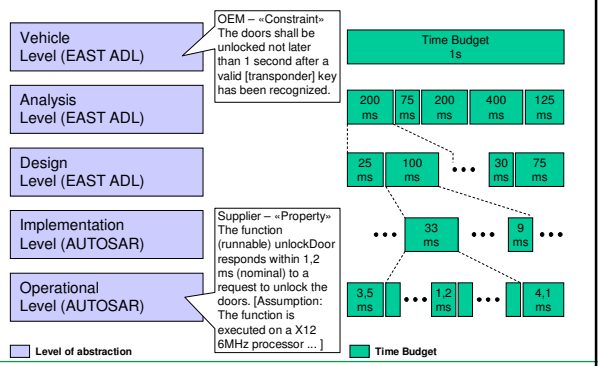
EAST-ADL
Levels of Abstraction



Objectives
Reflections on Timing Requirements and Properties

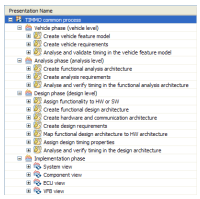


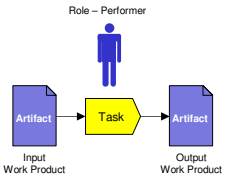
Objectives
Time Budgeting



Modeling Methodology
SPEM and Eclipse Process Framework (EPF) Composer

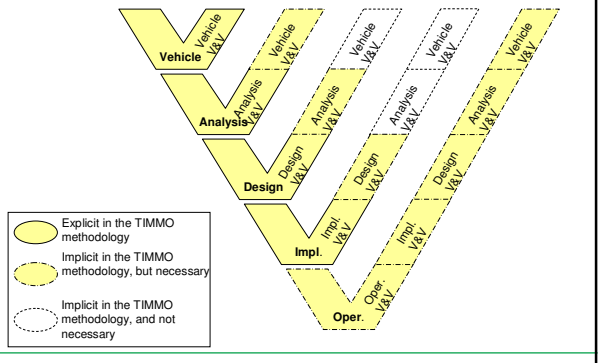
- Software Process Engineering Meta-Model
- Method content elements: Task, work product, and role
- Process elements: Phase, activity, task, milestone



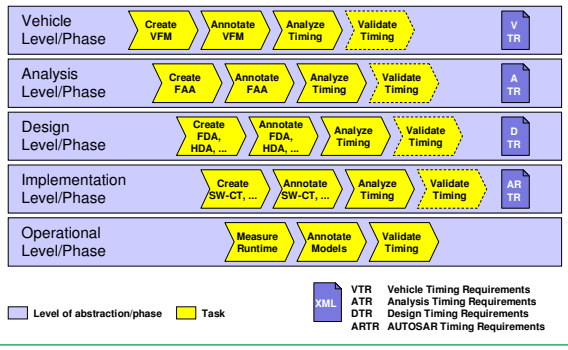


- Special care has been taken on the highly iterative and repeatable nature of the methodology on different levels:
  - Task
  - Sequence of tasks
  - Phases (Abstraction Levels)

TIMMO methodology



EAST-ADL
Methodology and Timing Artifacts – Simplified View

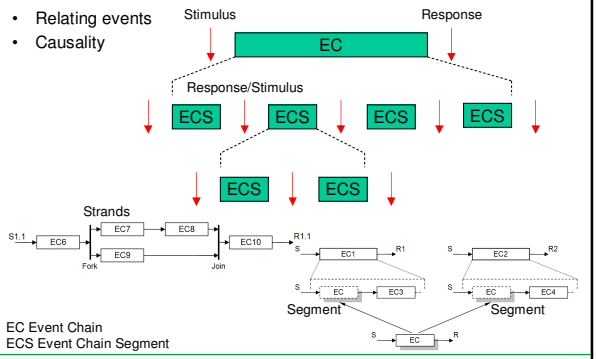


Events and Event Chains
Events

- Event
  - State or Change in State
  - Observable at specific locations in the system subject to analysis
- Event Model
  - Periodic
  - Sporadic
  - Pattern
  - Arbitrary

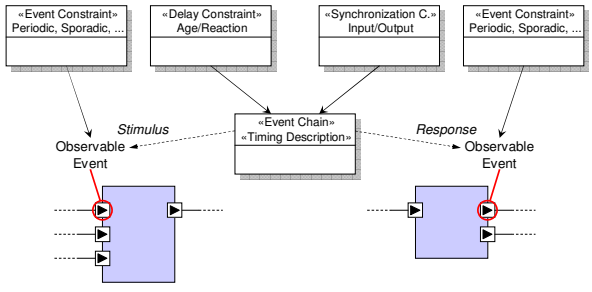
Events and Event Chains
Event Chains

- Relating events
- Causality

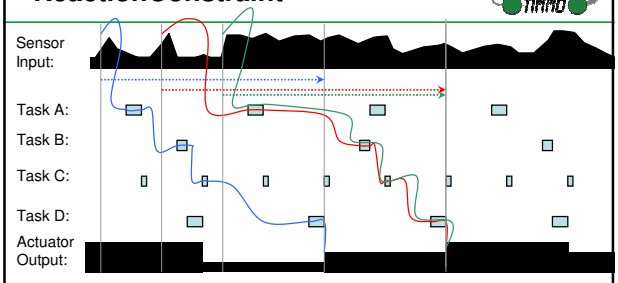


EC Event Chain
ECS Event Chain Segment

Events and Event Chains
Constraints and Description

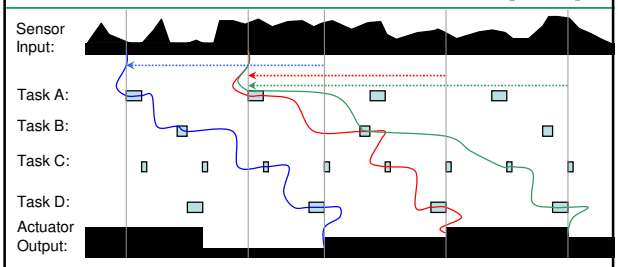


ReactionConstraint



For each change in (Sensor) Input:
Exact one well-defined Reaction Time path
How long does it take to get an effect on (Actuator) Output

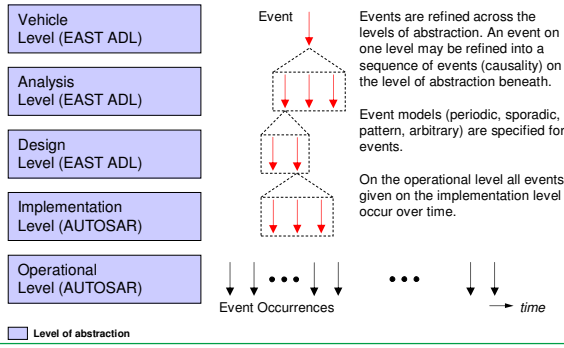
AgeTimingConstraint



For each change in (Actuator) Output:
Exact one well-defined Age path
How long ago did the corresponding sampling of (Sensor) Input occur

Events and Event Chains

EAST ADL Abstraction Levels, Events, and Timing



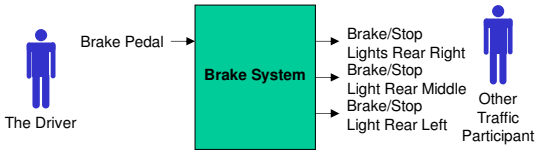
Events are refined across the levels of abstraction. An event on one level may be refined into a sequence of events (causality) on the level of abstraction beneath.

Event models (periodic, sporadic, pattern, arbitrary) are specified for events.

On the operational level all events given on the implementation level occur over time.

Example

Braking System – Boundaries

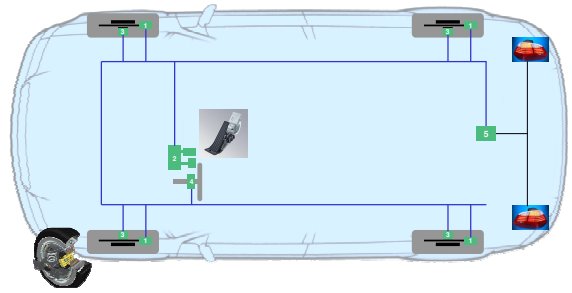


From the actor/user's (driver, other traffic participants) perspective the brake system consists of a brake pedal (sensor) and the stop lights (actuators). An assumption is that the brake actuators are part of the system called 'Brake System' but are not shown in the figure depicted above, due to the fact that these actuators are not directly visible to actors (driver and traffic participants).

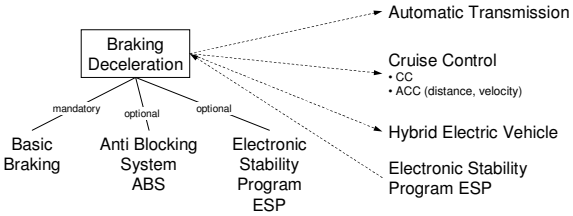
From a vehicle's point of view the Brake System simply is a box without any input/output arrows. So what is the relation with other vehicle functions? For example, the vehicle function Cruise Control also senses the brake pedal in order to temporarily turn off its operation when the driver pedals the brake pedal. In this case the brake pedal becomes a global visibility in the vehicle's system.

Example

Braking System – The Hardware View



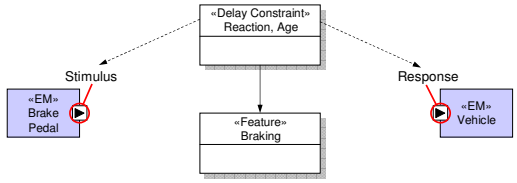
Example – Vehicle Level



- Timing requirement: The response time of the [feature] brake shall be less than 500 ms. [The driver shall make the experience that the breaks are taking into effect immediately after she/he presses the brake pedal.]
- The value of this requirement may change depending on other available features.

Example – Vehicle Level

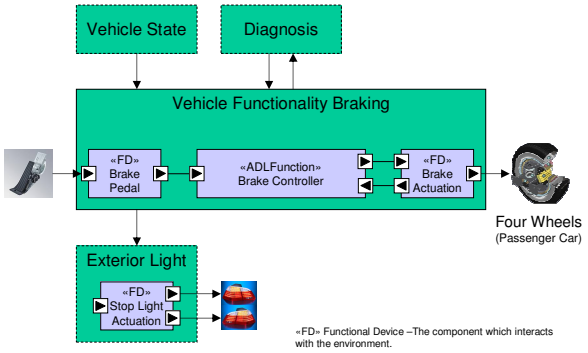
One proposal ... not yet approved



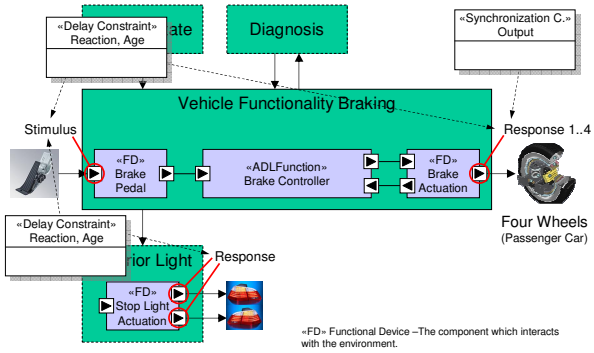
The environment model of the Brake Pedal describes how the brake pedal is pressed and which physical means are used to carry the information "how strong the brakes should be applied". The environment model of the Vehicle describes how the vehicle is slowed-down when the brake pedal is pressed.

On this level of abstraction the stimulus occurs sporadically ... no one is braking periodically!

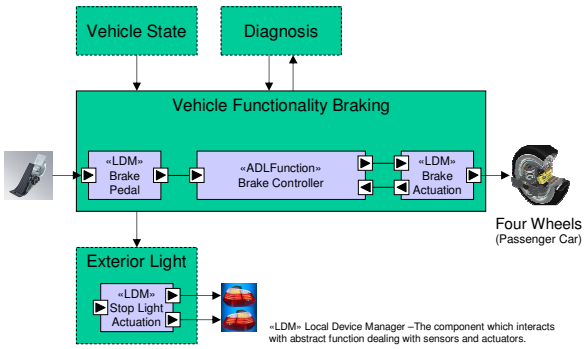
Example – Analysis Level



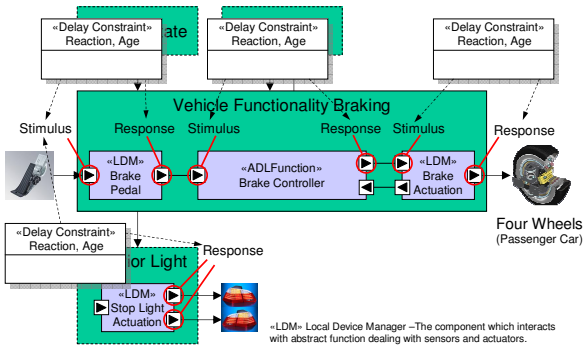
Example – Analysis Level



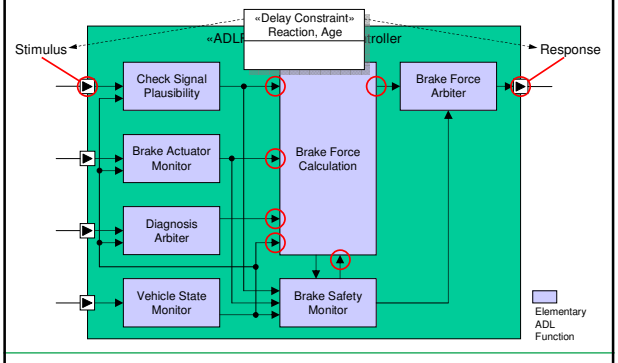
Example – Design Level
First Approximation based on Analysis Level



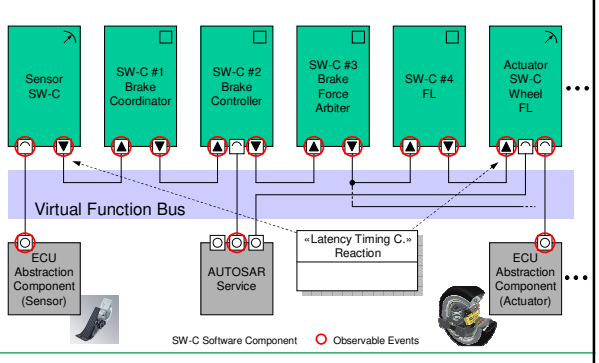
Example – Design Level
Time Budgets given from Analysis Level



Example – Design Level
Possible Design of the Brake Controller

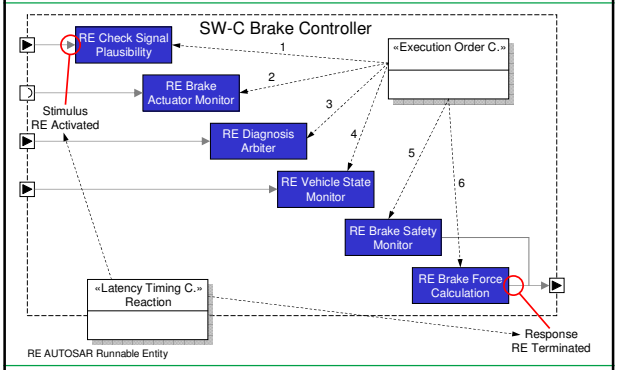


Example – Implementation Level
AUTOSAR Virtual Function Bus View



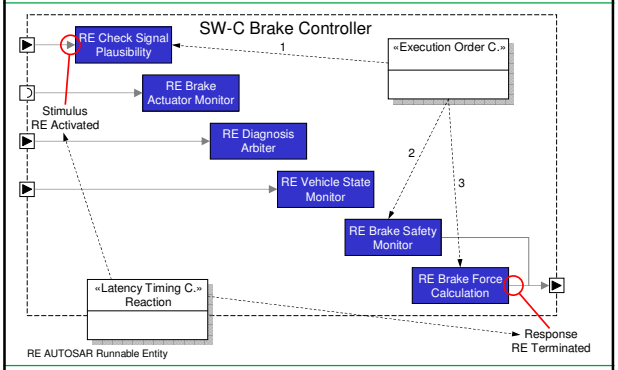
Example – Implementation Level

AUTOSAR Component View ... First alternative



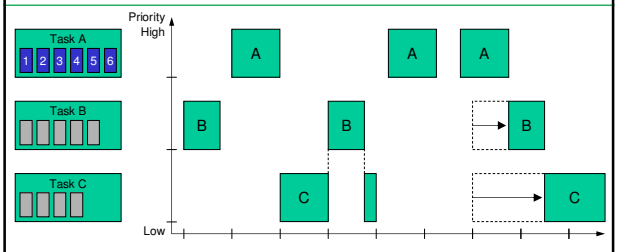
Example – Implementation Level

AUTOSAR Component View ... Second Alternative



Example – Implementation Level

AUTOSAR Configuration #1: Mapping REs to Tasks

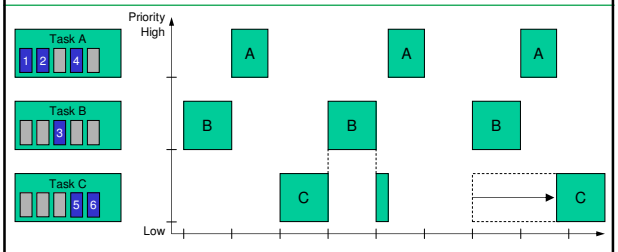


Task A Event Triggered (Sporadic)
Task B Event Triggered (Periodic)
Task C Time Triggered

Runnable Entities:
1 – Check Signal Plausibility
2 – Brake Actuator Monitor
3 – Diagnosis Arbiter
4 – Vehicle State Monitor
5 – Brake Safety Monitor
6 – Brake Force Calculation

Example – Implementation Level

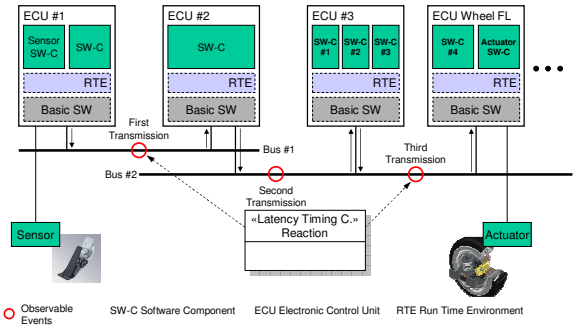
AUTOSAR Configuration #2: Mapping REs to Tasks



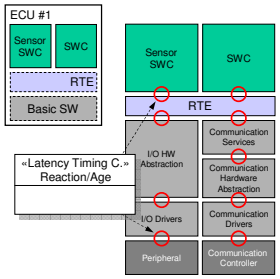
Task A Event Triggered (Sporadic)
Task B Time Triggered
Task C Time Triggered

Runnable Entities:
1 – Check Signal Plausibility
2 – Brake Actuator Monitor
3 – Diagnosis Arbiter
4 – Vehicle State Monitor
5 – Brake Safety Monitor
6 – Brake Force Calculation

Example – Implementation Level
AUTOSAR System View



Example – Implementation Level
AUTOSAR ECU View



- ECU View: Basic Software Module Entry Called, Basic Software Module Entry Returned
- Internal Behavior: Runnable Entity Activated, Runnable Entity Started, Runnable Entity Terminated, Basic Software Module Entity Activated, Basic Software Module Entity Started, Basic Software Module Entity Terminated
- Communication: Signal Sent To COM, Signal Available For RTE, IPDU Sent To Interface, IPDU Received by COM, Frame Queued for Transmission, Frame Transmitted on Bus, Frame Received by Interface

Questions and Discussion

Thank you very much for your attention!