Vehicle architectures for increasing autonomy

Elektronik i Fordon, 2012

Sagar Behere

25 April 2012

Kungliga Tekniska Högskolan

About me

- Doktorand - KTH/Embedded Control Systems (Supervisors: Martin Törngren, DeJiu Chen)
- Diesel engines, exhaust after-treatment, traction control
- Robotics, artificial intelligence, unmanned aerial vehicles

I will talk about

- Problems with current vehicle E/E architectures
- Solutions possible with autonomy based designs

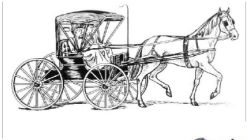
Research context

DFEA2020

Dependable and Flexible Electrical Architecture 2020

Enable green, safe and connected vehicles

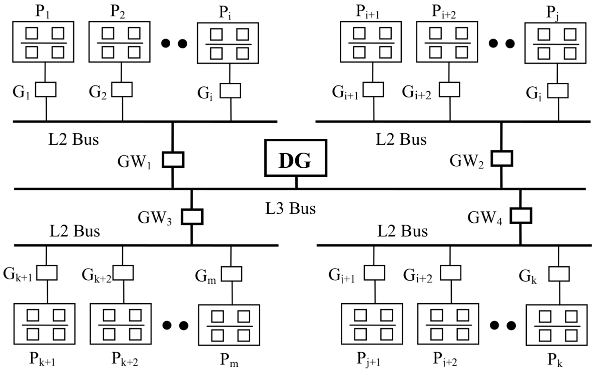
Perspective



Trends

- Autonomy
- Functional growth
- System complexity
- Networking - ubiquitous connectivity
- Stricter requirements
- Diversity of power sources

State of practice: Partitioning



(source: S.M. Mahmud, In-Vehicle Network Architecture for the Next-Generation Vehicles)

Necessary, but not sufficient!

Leading questions

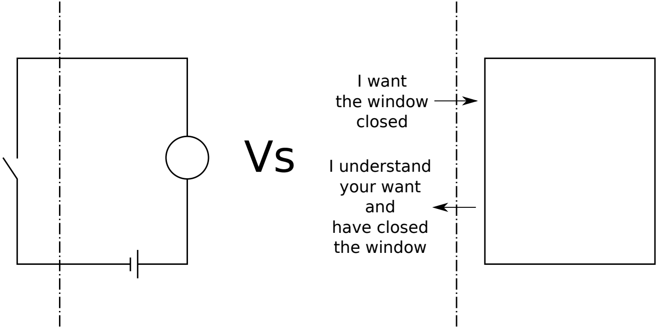
- How do we handle complexity?
- How much has the 'many nodes attached to a communication bus' principle evolved?
- Can we better design an architecture if we know it is destined for autonomy?

Proposition

Principles of autonomous systems design can be profitably applied to the design of automobile architectures

Autonomy

- **autos** = self ; **nomos** = law
- Having one's own laws. Self governing, capable of making its own decisions
- But what is the 'Self' in an engineering context?

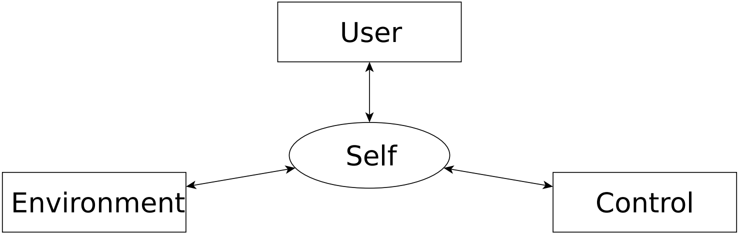


Intelligence

- The ability of a system to act appropriately in an uncertain environment
- Appropriate action is that which increases the probability of success
- Success is the achievement of behavioral sub-goals that support the system's ultimate goal
- The criteria of success and the system's ultimate goal may be defined external to the intelligent system

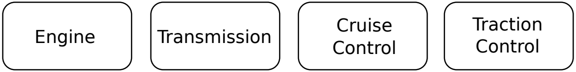
(source: J. Albus, Outline of a theory of intelligence)

Components of intelligent autonomy



The problem of implicit Selves

Current vehicle subsystems



- Capable of observing internal and external variables
- Contain model of own system being controlled
- No explicit Self that can understand
    - Overall purpose of the vehicle
    - Capabilities of the subsystems
    - How subsystems should interact to fulfill overall purpose

Knowledge is implicitly embedded in the brains of the designers.

The Self must be One

- One from the viewpoint of user interaction
- One from the viewpoint of design/construction
    - Extreme case: A Big Fat Computer

But what about

- distributed systems?
- composition of autonomous subsystems?

What I have said so far

- Autonomy, functional growth and complexity are trends
- Current vehicle architectures cannot support these trends
- The key problem is multiple, often conflicting, implicit Selves
- A unique Self, responsible for the entire vehicle, needs to be built right into the vehicle

Old wine in a new bottle?

- **Yes**. Existing subsystems aren't being tossed out
- **No**. We need a reasoned and systematic basis for architecting intelligent, autonomous vehicles that can
  - assimilate current practices
  - impose a logic on what we have been doing
  - bring out flaws and gaps in current practices and provide means to eliminate them

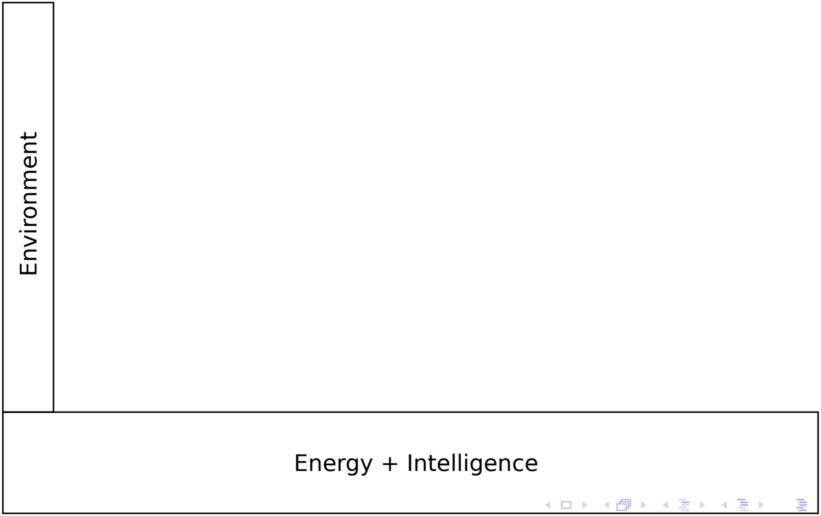
Inspiration

- Lot of ongoing research
- End goals markedly similar
- Lower historical 'baggage'

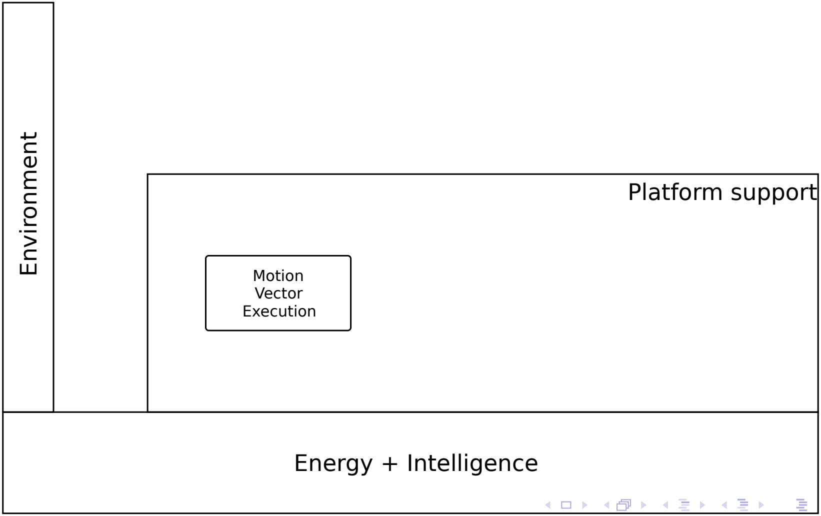
Exercise: Architecting a robotic car

Energy + Intelligence

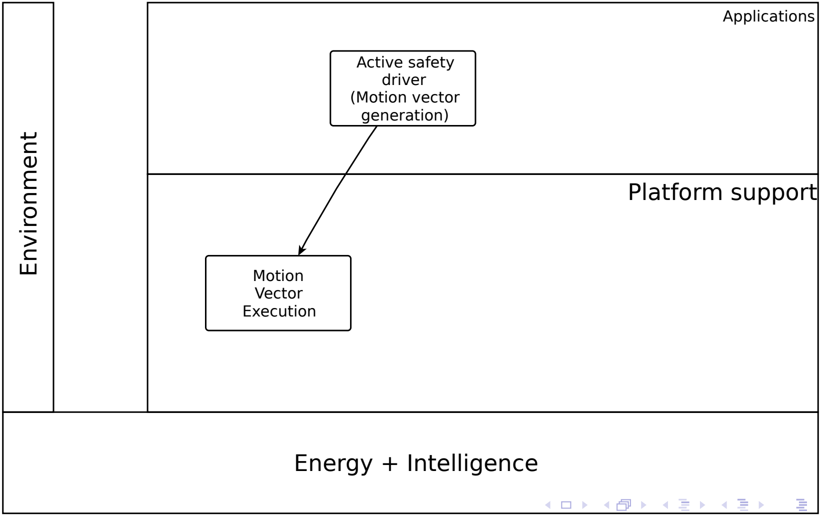
Exercise: Architecting a robotic car



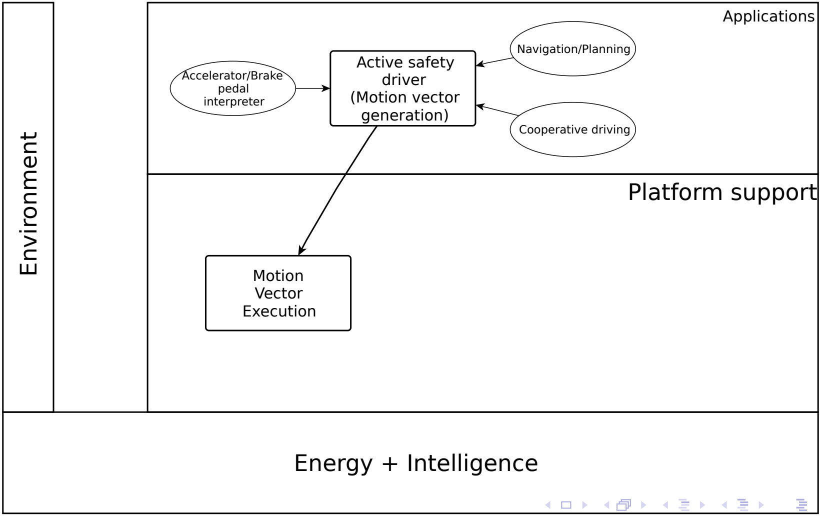
Exercise: Architecting a robotic car



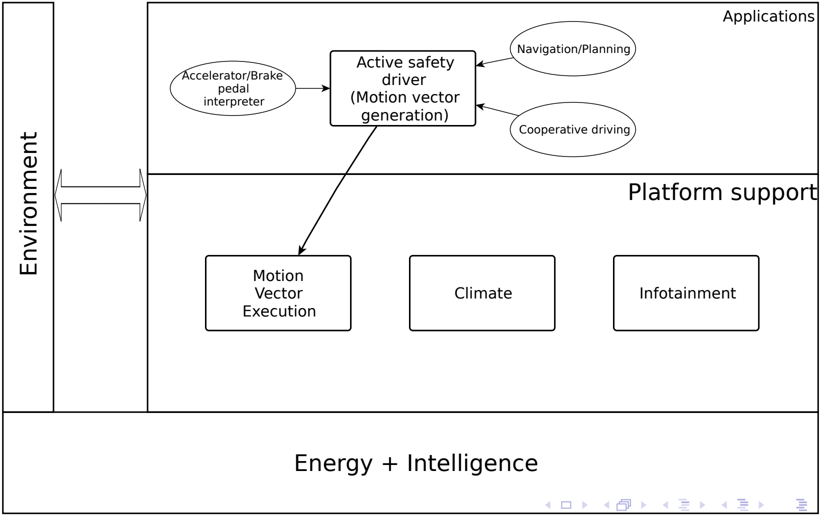
Exercise: Architecting a robotic car



Exercise: Architecting a robotic car



Exercise: Architecting a robotic car



Characteristics

- Clear direction towards autonomy and autonomous features
- Everything is software (device drivers + applications) and
- Software can be structured and partitioned with greater flexibility
- Management of functional growth and complexity via big picture thinking
    - Every function has a place and interface in the autonomous scheme of things

Reflections

- Automotive systems have traditionally grown "bottom up"
- A "top down" approach combining functional, hardware and other views is needed
- Separation of concerns achieved by two means
  - Modularization
  - Abstraction
- Hierarchies embody both means, but still necessary to decompose them properly

In summary

- It is time to stop thinking of a vehicle as an increasingly mechanized/computerized horse drawn carriage
- Complex systems can't be simplified by using a magic architectural wand, but
- Certain architectures can accommodate complexity better than others
- Thinking of the car as an autonomous robot may show the way out

Questions?

behere@kth.se