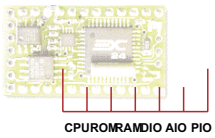
How to REALLY Customize Your Car!!

– A Nascent RoSES Customization Manager

RoSES
Robust Self-Configuring Embedded Systems

William Nace

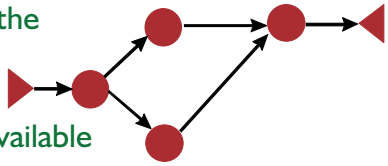
Where, oh where, should my software run?



Given:

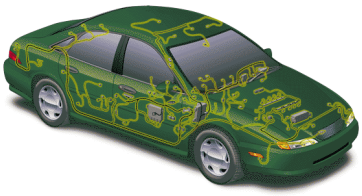
- A collection of microcontrollers, each with a vector of discrete resources.
- A software flow graph, where functionality nodes have resource requirements & arcs have dataflow requirements.

Do:

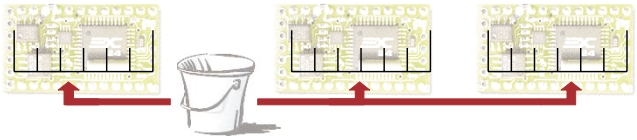


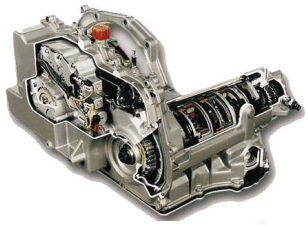
- Pack the nodes -- starting with "largest" -- onto the microcontroller that will minimize network use.
- Watch for size or other constraints!!
- Continue until all nodes pack or they overflow available microcontrollers.

How many embedded processors do I need?



- Start with minimal set of small microcontrollers.
- Keep packing tasks until failure.
- At each failure, decide to either grow a microcontroller's specification or add another processor.
- Make decision based on cost model and effect on network bandwidth.





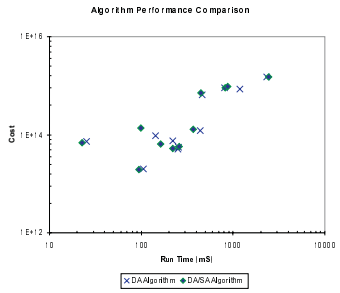
What if I want certain sensor & actuator software to co-exist? Or not?

Cluster during a pre-packing phase.

Exclude when checking packing constraints.

BONUS - Algorithm runs faster!

Data Set	Nodes	#S/A	Avg #Grps	Avg Grp Size
Ran01	100	20	2	4.9
Ran02	100	20	2	6.9
Ran05	50	28	4	6.7
Ran06	200	196	7	5.6
Ran07	50	8	2	3.3
Ran08	75	32	4	4.1
Ran09	90	73	7	5.7
Ran10	25	5	2	2.0
Ran11	100	99	7	2.1
Syn01	8	5	2	2.0
Traction	160	117	7	4.9



Where will this research go from here?

Customization manager for RoSES will:

- Optimize functionality system-wide for available hardware
- Choose algorithms/adapters for installation on sensors & actuators
- Abide by real-time scheduling constraints, both in the CPU & network

GOALS:

- Graceful degradation
- Graceful upgrade
- Product family architecture design
- Logistic benefits
  - Replacement with non-exact spares
  - Reduced need for legacy parts

GM

Carnegie Mellon

Electrical & Computer ENGINEERING

Institute for Complex Engineered Systems