Extensible and Scalable Time Triggered Scheduling for Automotive Applications

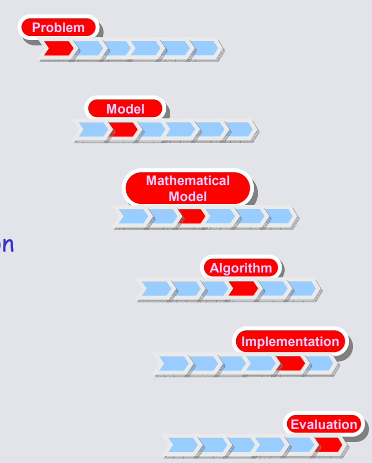
Wei Zheng

Advisor: Professor Alberto Sangiovanni-Vincentelli

Chess Review
May 11, 2005
Berkeley, CA

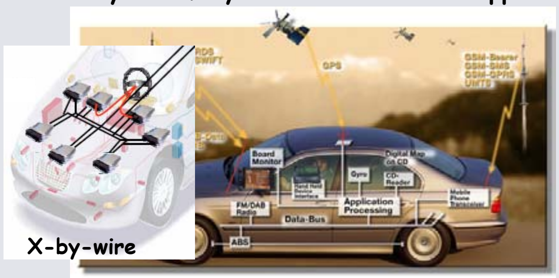
Agenda

- **Motivation**
- **Problem Statement**
- **Previous Work**
- **Investigative Approach**
  - Metric Definition
  - Mathematical Formulation
  - Multi-Objective Cost Function
- **Case Study**
  - System Description
  - Cost Function Evaluation
  - Metrics Evaluation
- **Conclusion**
- **Future Work**



Project Motivation

- Hard Real-time Embedded Systems are ubiquitously used today in safety critical commercial applications

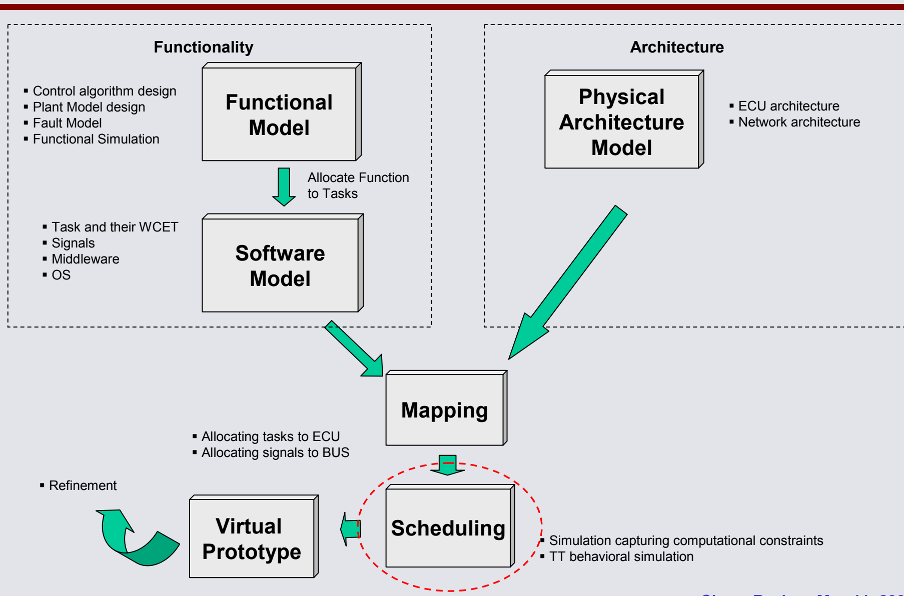


Power Transmission Unit
- 6-lines per day
- 3000 ppm residential defects
- 5 months validation time
FABIO ROMEO, Magneti- Marelli
DAC, Las Vegas, June 20th, 2001

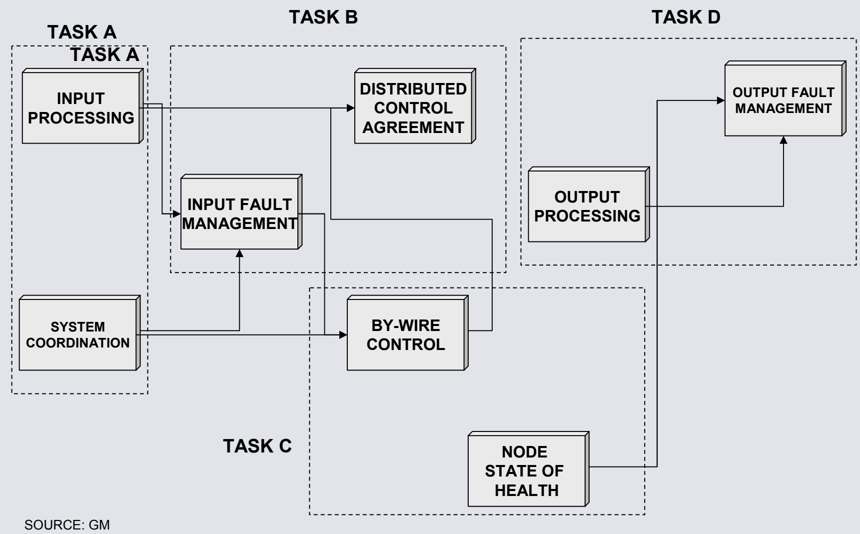
- Verification of complex systems is time and resource intensive

- For fast time-to-market ➔ Extensible and Scalable systems

Design Flow

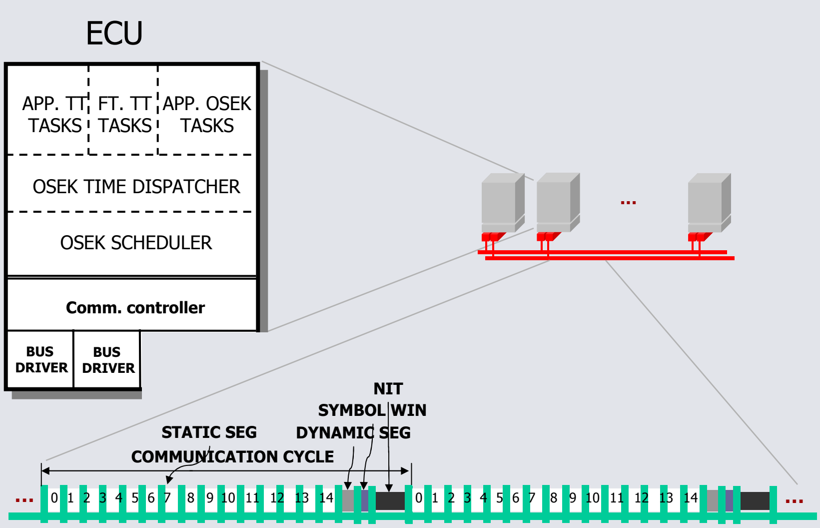


Functionality: Allocate Function to Task



SOURCE: GM

System Architecture

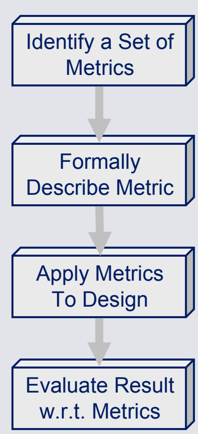


Agenda

- **Project Motivation**
- **Problem Statement**
- **Previous Work**
- **Investigative Approach**
  - Metric Definition
  - Mathematical Formulation
  - Multi-Objective Cost Function
- **Case Study**
  - System Description
  - Cost Function Evaluation
  - Metrics Evaluation
- **Conclusion**
- **Future Work**

Problem Statement

- **Identify a set of metrics to capture extensibility and scalability**
- **Apply the set of metrics in a design**
- **Evaluate the effectiveness of the set of metrics**
- **Specifically, we want to:**
  - Study a hard real time embedded systems in the automotive domain
  - Focus on the **scheduling** aspect of system design
  - Characterize extensibility and scalability in scheduling
  - Apply the set of metrics in a scheduling algorithm
  - Evaluate the effectiveness of the approach with industrial case study



Agenda

- Project Motivation
- Problem Statement
- Previous Work
- Investigative Approach
  - Metric Definition
  - Mathematical Formulation
  - Multi-Objective Cost Function
- Case Study
  - System Description
  - Cost Function Evaluation
  - Metrics Evaluation
- Conclusion
- Future Work

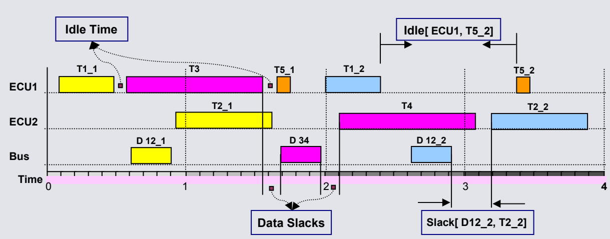
Previous Work

- Static cyclic scheduling has been extensively researched
- Classical scheduling theory use metrics such as
  - Minimizing sum of completion time
  - Minimizing schedule length
  - Minimizing resource
- For real time systems, deadline is added as a constraint
  - Emphasis shifted to finding feasible solutions while
    - Minimizing end-to-end delay
    - Minimizing communication cost
- Closest problem concept comes from Paul Pop, et al
- Closest problem formulation comes from Armin Bender, et al

Previous Work

- **Paul Pop, et al, wrote about incremental design**
  - Use list scheduling approach to obtain a valid schedule
  - Use a heuristic to distribute slack in the system
  - Missing several important components
    - Preemption is not considered
    - Resulting schedule is not suitable for future task with urgent deadline
    - Message slack is not distributed
    - Extensibility is not considered
- **Armin Bender, et al, used mathematical programming for mapping and scheduling**
  - Work is motivated by software-hardware co-design
  - Objective is to obtain schedule feasibility while
    - Maximizing Performance
    - Minimizing resource

Research Direction



- **Focus on optimally utilize redundancies in schedules for extensibility and scalability**
  - Idle time and slacks are traditionally incorporated in hard real time embedded systems schedules to increase system robustness
- **We should utilize these redundancies to:**
  - Tolerate incremental design changes
  - Accommodate new tasks to be added in future product updates

Agenda

- **Project Motivation**
- **Problem Statement**
- **Previous Work**
- **Investigative Approach**
  - Metric Definition
  - Mathematical Formulation
  - Multi-Objective Cost Function
- **Case Study**
  - System Description
  - Cost Function Evaluation
  - Metrics Evaluation
- **Conclusion**
- **Future Work**

Capture the Metrics

Model

Extensibility		Scalability
• Tolerate changes of Task WCET • Tolerate changes of Data WCTT	**Motivation**	• Accommodate NEW tasks by statically scheduling them on a legacy system
• Maintain Bus Schedule • Maintain non-involved ECU schedules • Maintain involved ECU schedules without reconfiguration	**Implementation**	• Provide blocks of computation time for future computation intensive tasks • Provide porosity in schedules to allow for future tasks with tight deadlines

• Message left & Right slack
  • Max Sum of all slacks
  • Min Variance of all slacks
	Approach	• ECU idle time distribution
		• Bus idle time distribution
		  • Evenly distribute all idle time

Applying the Metrics

Mathematical Model

- **Develop a formal representation of the problem using mathematical programming and solve it using existing solver**
  - Modeling Language: **AMPL**
  - *Advantage:* obtain optimal solution w.r.t. cost function
  - *Disadvantage:* computationally intensive suitable only for moderately sized problems
- **Assumptions:**
  - Hard real time deadlines
  - Statically scheduled tasks with data dependency
  - Distributed and heterogeneous multi-processor architecture
  - Time triggered bus with TDMA protocol
  - Preemption allowed on ECUs with no level limits
  - Multi-rate task support with adaptive task graph expansion
  - Fixed task allocation with no task migration

Mathematical Formulation 1

Mathematical Model

Notations

- The set of Tasks $T = \{t_i \mid i = 1,\ldots,m\}$
- The set of ECU $E = \{e_i \mid i = 1,\ldots,n\}$
- The set of task pair with data dependency running on the same ECU
  $\sigma = \{(t_i, t_j) \mid t_i \in T, t_j \in T, t_i < t_j, a_{i,k} = a_{j,k}\}$
- The set of task pair with data dependency running on different ECU
  $\varpi = \{(t_i, t_j) \mid t_i \in T, t_j \in T, t_i < t_j, a_{i,k} + a_{j,k} \le 1\}$
- The set of task non-reachable task pair running on the same ECU
  $\theta = \{(t_i, t_j) \mid t_i \in T, t_j \in T, a_{i,k} = a_{j,k}, t_i <> t_j\}$
- The set of task pair running on the same ECU
  $\pi = \{(t_i, t_j) \mid t_i \in T, t_j \in T, a_{i,k} = a_{j,k}, k \in E\}$
- The set of task allocation for ECU
  $\lambda = \{\kappa_j \mid j \in E\}, \kappa_j = \{t_i \mid a_{i,j} = 1\}$

Mathematical Formulation 2

Mathematical Model

Parameters and Variables

- Mapping from Tasks to ECUs $M = T \times E = \{a_{i,j} \mid i = 1,\ldots,m;\ j = 1,\ldots,n\}$

$$a_{i,j} = \begin{cases} 1 & \text{if task i is mapped to ECU j} \\ 0 & \text{otherwise} \end{cases}$$

- Task and Message

Task: 6-tuple parameter variable Vector $\tau : \langle e, r, p, c, s, f \rangle$	
$e_i, i \in T$	WCET
$r_i, i \in T$	Release Time
$p_i, i \in T$	Period
$c_i, i \in T$	Idle time
$s_i, i \in T$	Starting time
$f_i, i \in T$	Finishing time

Message: 5-tuple parameter variable Vector $\varsigma : \langle mt, sl, sr, ms, mf \rangle$	
$mt_{i,j}, (i,j) \in \varpi$	WCTT
$sl_{i,j}, (i,j) \in \varpi$	Left Slack
$rl_{i,j}, (i,j) \in \varpi$	Right Slack
$ms_{i,j}, (i,j) \in \varpi$	Starting time
$mf_{i,j}, (i,j) \in \varpi$	Finishing time

Mathematical Formulation 3

Mathematical Model

Parameters and Variables (continue)

- Idle time and Integer Variables

c_i^{idle} Idle time for task i, if i is not the first task in of its running ECU

c_k^{first} First idle time for each ECU k

$c_k^{after-last}$ Idle time for ECU k before the super period

$$y_{i,j} = \begin{cases} 0 & \text{if the starting time of task i precede the starting time of task j} \\ 1 & \text{otherwise} \end{cases} \qquad (i,j) \in \theta$$

$$p_{i,j} = \begin{cases} 0 & \text{if task i is not preempted by task j} \\ 1 & \text{otherwise} \end{cases} \qquad (i,j) \in \theta$$

$$z_{i,j,k,l} = \begin{cases} 0 & \text{if data transmitted from task i to task j precedes data transmitted from task k to l} \\ 1 & \text{otherwise} \end{cases} \qquad (i,j) \in \varpi,\ (k,l) \in \varpi$$

Mathematical Formulation 4

Mathematical Model

- Subject to the following constraints:
  - Release and Deadline Constraints
  - Execution Time/Transmission Constraints
  - Precedence Constraints
  - Non-negative and Integer Constraints

$$r_i \le s_i, i \in T \qquad f_i \le d_i, i \in T$$

$$f_i = s_i + e_i + \sum_{(i,j)\in\theta} p_{i,j} \times e_j, i \in T, j \in T$$

$$ms_{i,j} + mt_{i,j} = mf_{i,j}, (i,j) \in \varpi$$

$$f_i \le s_j, (i,j) \in \sigma$$

$$f_i <= ms_{i,j}, (i,j) \in \varpi$$

$$ms_{i,j} + mt_{i,j} \le s_j, (i,j) \in \varpi$$

$$s_i \le s_j + y_{i,j} \times \Lambda, (i,j) \in \theta$$

$$s_j \le s_i + (1 - y_{i,j}) \times \Lambda, (i,j) \in \theta$$

$$y_{i,j} + y_{j,i} = 1, (i,j) \in \theta, (j,i) \in \theta$$

$$s_i \ge 0, f_i \ge 0, i \in T$$

$$ms_{i,j} \ge 0, mf_{i,j} \ge 0, (i,j) \in \varpi$$

$$y_{i,j}, p_{i,j} = binary, (i,j) \in \theta$$

$$z_{i,j,k,l} = binary$$

$$(i,j) \in \varpi, (k,l) \in \varpi, i \ne k, or, j \ne l$$

Mathematical Formulation 5

Mathematical Model

- Constraints (continued):
  - Mutual exclusion constraints

$$f_i \le s_j + y_{i,j} \times \Lambda + p_{i,j} \times \Lambda, (i,j) \in \theta$$

$$f_j \le f_i + y_{i,j} \times \Lambda + (1 - p_{i,j}) \times \Lambda, (i,j) \in \theta$$

$$p_{i,j} + p_{j,i} \le 1, (i,j) \in \theta, (j,i) \in \theta$$

$$p_{i,j} \le 1 - y_{i,j}, (i,j) \in \theta$$

$$ms_{i,j} + mt_{i,j} \le ms_{m,n} + z_{i,j,m,n} \times \Lambda$$

$$(i,j) \in \varpi, (m,n) \in \varpi, i \ne m, or, j \ne n$$

$$ms_{m,n} + mt_{m,n} \le ms_{i,j} + (1 - z_{i,j,m,n}) \times \Lambda$$

$$(i,j) \in \varpi, (m,n) \in \varpi, i \ne m, or, j \ne n$$

  - Idle time constraints

$$c_j^{idle} \le (s_j - f_i) + y_{i,j} \times \Lambda + p_{i,j} \times \Lambda, (i,j) \in \pi$$

$$c_i^{idle} \le s_i + y_{i,j} \times \Lambda, (i,j) \in \pi$$

$$c_j^{idle} \le 0 + y_{i,j} \times \Lambda + (1 - p_{i,j}) \times \Lambda, (i,j) \in \pi$$

$$c_k^{after\_last} \le P - f_i, k \in E, i \in \kappa_k$$

$$\sum_{j\in\kappa_k} c_j^{idle} + c_k^{after\_last} = P - \sum_{j\in\kappa_k} e_j, \forall k \in E$$

Agenda

- **Project Motivation**
- **Problem Statement**
- **Previous Work**
- **Investigative Approach**
  - Metric Definition
  - Mathematical Formulation
  - Multi-Objective Cost Function
- **Case Study**
  - System Description
  - Cost Function Evaluation
  - Metrics Evaluation
- **Conclusion**
- **Future Work**

Multiple Objective Cost Function

Algorithm

- **Extensibility**

$$\max R = \sum_{(i,j)\in\varpi}((ms_{i,j} - f_i) + (s_j - mf_{i,j}))$$

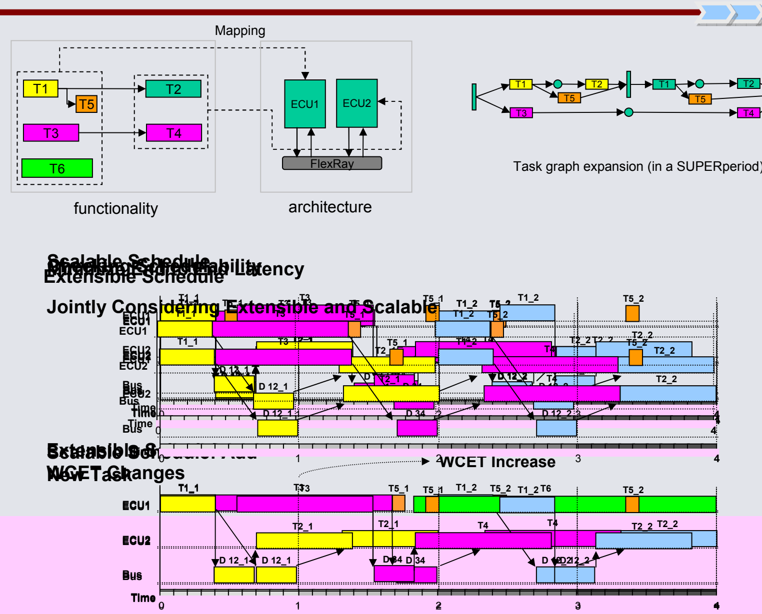
- **Scalability**

$$\min S = \sum_{j\in E}\sum_{i\in T} a_{i,j} \times (c_i^{idle} - average_j)^2 + \sum_{j\in E}(c_j^{after\_last} - average_j)^2$$

- **Jointly Extensibility & Scalability**

$$\min RS = K_1 \times (-R) + K_2 \times S$$

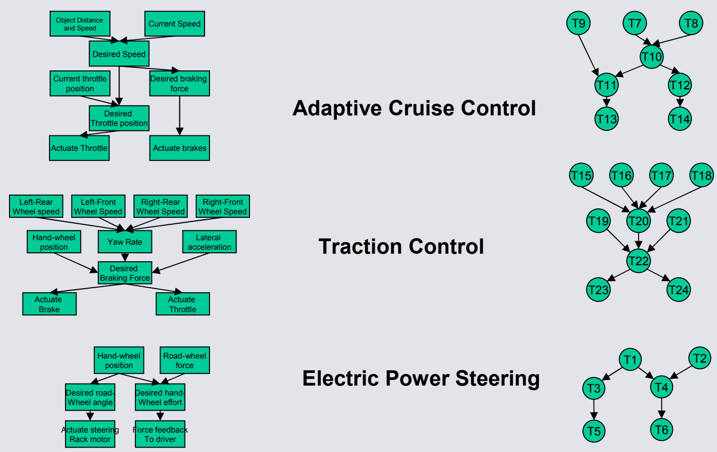
Extensibility and Scalability of Time Triggered Scheduling



Agenda

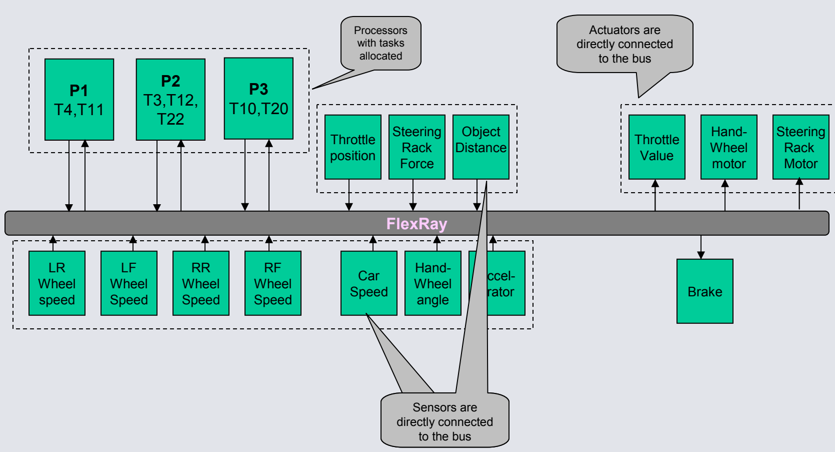
- Project Motivation
- Problem Statement
- Previous Work
- Investigative Approach
  - Metric Definition
  - Mathematical Formulation
  - Multi-Objective Cost Function
- **Case Study**
  - System Description
  - Cost Function Evaluation
  - Metrics Evaluation
- Summary
- Conclusion
- Future Work

Advanced Automotive Control Application



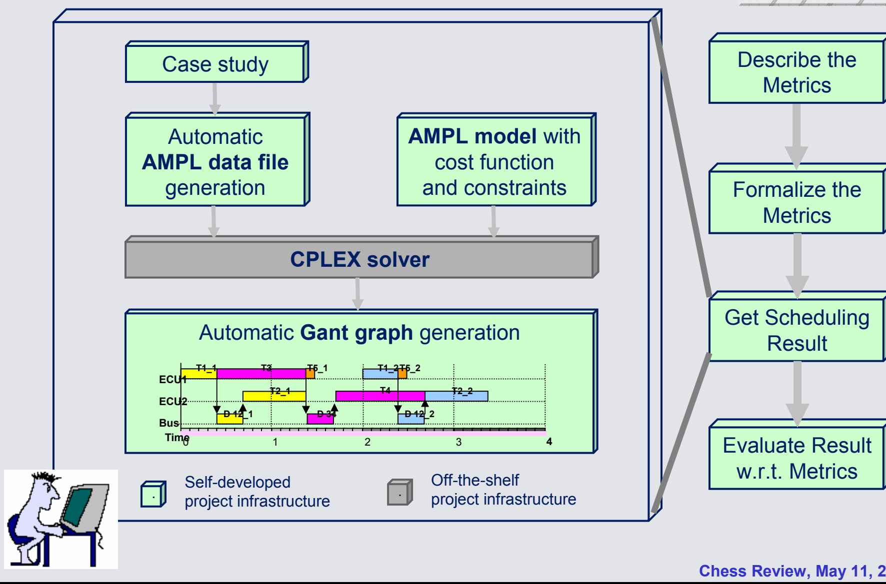
Applications and corresponding task graph representations

Architecture and Task Allocation



Implementation Infrastructure

Implementation



Agenda

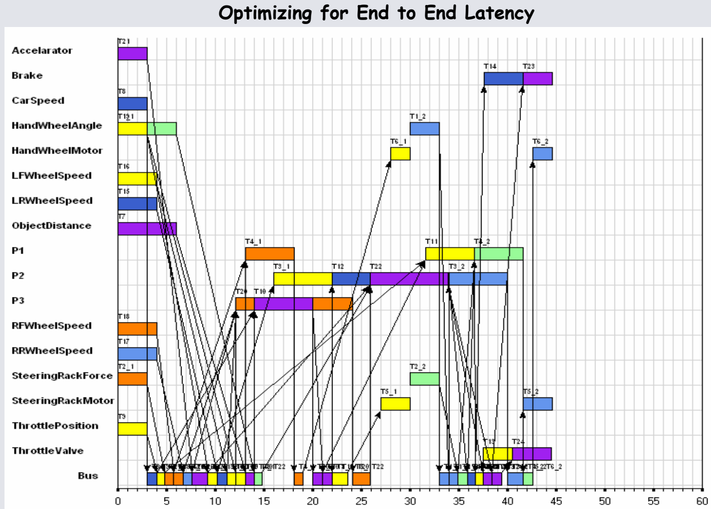
- **Project Motivation**
- **Problem Statement**
- **Previous Work**
- **Investigative Approach**
  - Metric Definition
  - Mathematical Formulation
  - Multi-Objective Cost Function
- **Case Study**
  - System Description
  - Cost Function Evaluation
  - Metrics Evaluation
- **Summary**
- **Conclusion**
- **Future Work**

Cost Function Evaluation

- **Multi-objective cost function is an abstraction**
  - Mathematical programming formulation has limited semantics
  - Extensibility and scalability metrics are too complex
  - Described in full, the cost function would be too computationally expensive
- **Must determine if the cost function abstraction effectively represents the metrics**
  - Use the results of CPLEX solver
  - Extract real slack and idle time distributions based on precise definition of the metrics
  - Compare results with the schedule without extensibility and scalability optimization

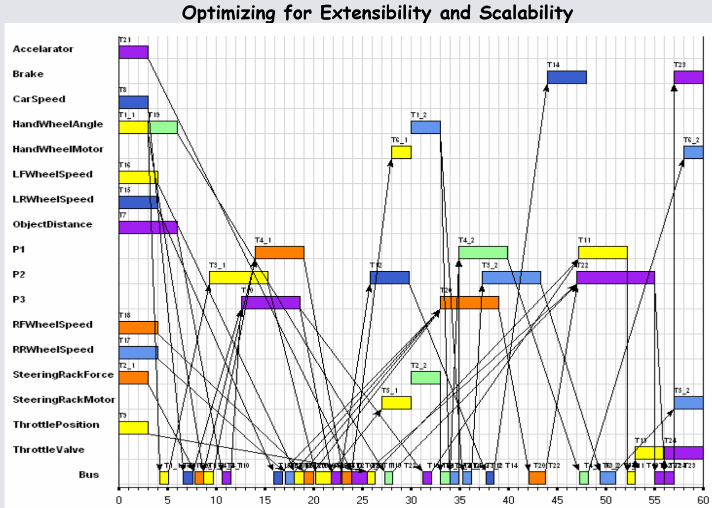
Traditional Scheduling Result

Optimizing for End to End Latency



Optimized Scheduling Result

Optimizing for Extensibility and Scalability

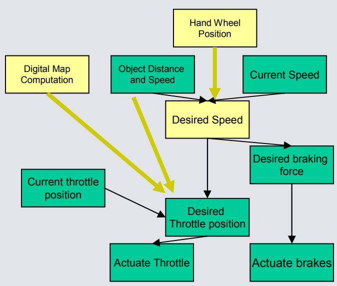


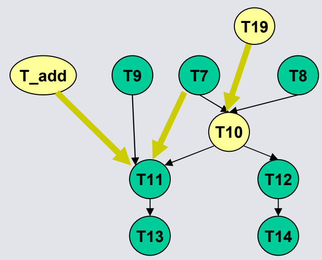
Metrics Evaluation

- **Our set of metrics is one abstraction of the extensibility and scalability concept**
- **Must determine if our metrics effectively handles incremental design changes**
- **Incremental Design Scenario:** Basic ACC → Stop-N-Go ACC
  - Addition of a new Adaptive Cruise Control feature
  - Predict desired speed based on:
    - Digital map information
    - Forward looking vision sensor
  - Requires addition of tasks and messages
  - Some existing tasks will need more computation time

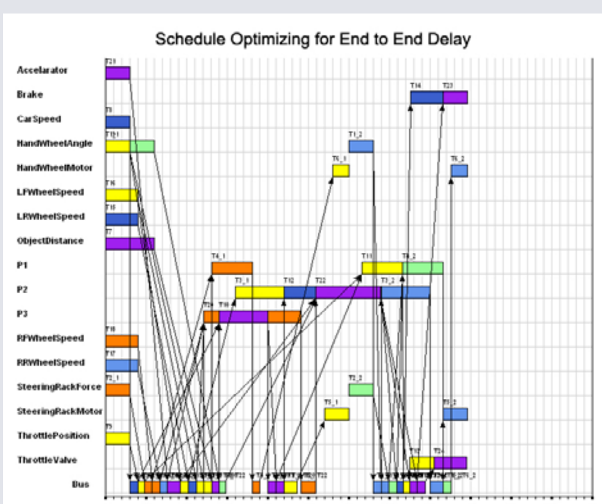
Adaptive Cruise Control

- **Incremental Design Changes:**
  - Add new ***Digital Map Computation*** task on P1
  - More complex algorithm in T10 (***Desired Speed Control***)
  - ***Desired Speed Control*** requires new input from ***Hand Wheel Sensor***
  - ***Desired Throttle Control*** requires new input from ***Forward Vision Sensor***





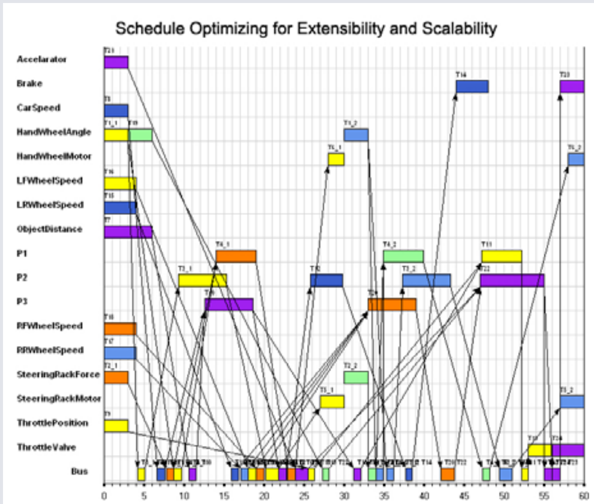
Traditional Schedule



In Tradition Schedule:

Incremental changes impossible without full rescheduling

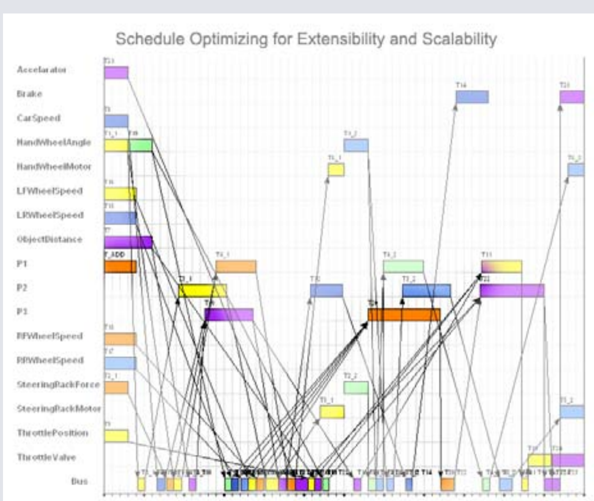
Optimized Schedule



In Tradition Schedule:
Incremental changes impossible without full rescheduling

In Optimized Schedule:
A lot more porosity to accommodate new tasks and messages

Optimized Schedule



In Tradition Schedule:
Incremental changes impossible without full rescheduling

In Optimized Schedule:
A lot more porosity to accommodate new tasks and messages

New functions added:
Without disturbing legacy schedules

Agenda

- Project Motivation
- Problem Statement
- Previous Work
- Investigative Approach
  - Metric Definition
  - Mathematical Formulation
  - Multi-Objective Cost Function
- Case Study
  - System Description
  - Cost Function Evaluation
  - Metrics Evaluation
- Conclusion
- Future Work

Conclusion

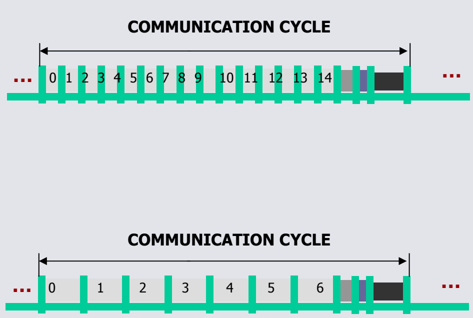
- Successfully captured extensibility and scalability metrics
- Recognized implications in accelerating time-to-market of embedded system development
  - Reduce re-verification burden in incremental design flow
  - No increase in resource requirements
- Formulated the scheduling problem as a mathematical programming problem
- Constructed multi-object cost functions abstracted from the metrics
- The cost function is shown to be effective for the metrics
- The metrics is shown to be effective in industry case study

Agenda

- **Project Motivation**
- **Problem Statement**
- **Previous Work**
- **Investigative Approach**
  - Metric Definition
  - Mathematical Formulation
  - Multi-Objective Cost Function
- **Case Study**
  - System Description
  - Cost Function Evaluation
  - Metrics Evaluation
- **Conclusion**
- **Future Work**

Future Work

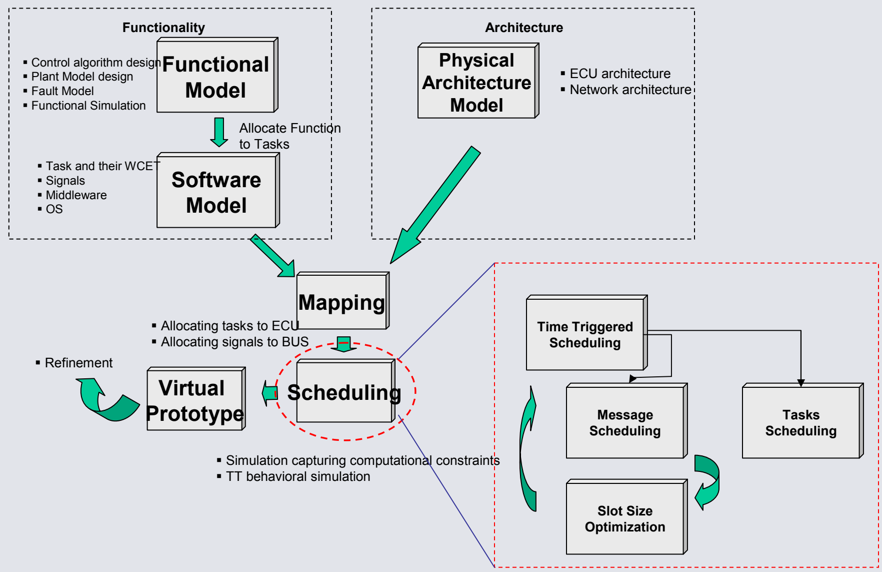
- Protocol Comparison
  - FlexRay Vs. TTP
- Slot Size Optimization



Slot Size Exploration

- Read/Write
- Message Frame Packing
- Buffer Requirement
- Fragmentation

Future Work



Reference

- Paul Pop: Analysis and Synthesis of Communication-Intensive Heterogeneous Real-Time Systems. Ph. D. Thesis No. 833, Dept. of Computer and Information Science, Linköping University, June 2003
- H. Kopetz et al., Real-Time Systems-Design Principles for Distributed Embedded Applications, Kluwer Academic Publishers, 1997
- N. Kandasamy, J. P. Hayes, B. T. Murray. Dependable Communication Synthesis for Distributed Embedded Systems.
- Liu, Layland, Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment, J. ACM 20, p. 46-61, 1973
- Devillers, Goossens, Liu and Layland's schedulability test revisited, Information Processing Letters, p 157-161, 2000
- Bini, Gio. Buttazzo, Giu. Buttazzo, Rate Monotonic Analysis : The Hyperbolic Bound, p 933-943, 2003
- Pradyumna K. Mishra and Sanjeev M. Naik. Distributed Control System Development for FlexRay-Based Systems
- A. Bender, "Design of an Optimal Loosely Coupled Heterogeneous Multiprocessor System," in Proceedings of Electronic Design and Test Conference, pages 275-281, 1996