Intelligent Hybrid Vehicle Management Systems

Hubert H. Chin and Ayat A. Jafari
Electrical Engineering and Computer Science Department
New York Institute of Technology
Old Westbury, New York 11568

Abstract- This paper addresses system design and integration challenges involved in meeting the requirements for coordinated deployment of Intelligent Hybrid Vehicle Systems. A hybrid vehicle can run purely on a gas engine, purely on an electric motor, or a combination of both. Advanced hybrid vehicles are designed to use a vehicle management system that integrates many subsystem functions into one system. However, many recent hybrid vehicles do not focus on the utilization of decision support techniques for the implementation of system design. The decision support design emphasizes a driver-oriented and automatic-oriented drive-by-wire vehicle system. The top-down functional decomposition and bottom-up physical integration design methodologies help to create an augmented functional analysis in the device level software. The significant portions are devoted to introducing the design stages and establishing the hybrid vehicle management system.

Keywords- - Intelligetn Systems, Hybrid Vehicle, System Design, Vehicle Management System, Top-down and Bottom-up Design Methods.

I. INTRODUCTION

The Intelligent Hybrid Vehicle Management System (IHIHVMS) is intended to use decision support techniques to integrate the existing hybrid vehicle subsystems into a system which will provide the driver with optimal ways to: handle vehicle operations; make critical decisions; and achieve energy conservation. Hybrid vehicles utilize two or more power sources in the drive train. The gasoline-electric hybrid vehicles[1] are currently available on the market. Decision support techniques are interactive computer software intended to identify problems and make decisions which are part of Driver Associated (DA) technology and based upon modern decision theory[2]. The DA is an intelligent vehicle project[3] which provides a digital assistant for the driver. The system constantly updates a traffic data file, and alerts the driver when any hazardous road conditions appear ahead. It can modify the cruise control setting based on its knowledge base of the current road conditions. In some cases, the driver may not even realize that assistance was provided.

The DA also has the ability to constantly monitor the relationship between the vehicle, the lane, and the heading with respect to the road. This is achieved using a video camera system which is capable of detecting, classifying, and assessing potential collision threats in the vehicle's path. A light detection and ranging[4] (LIDAR) device is a laser range finder reflected by a rotating mirror on the roof of the vehicle. The laser scans around the scene being digitized in the middle section, gathering distance measurements at specified angle intervals at the bottom. Four standard automotive radar sensors help determine the positions of distant objects. The Position Estimator is a sensor mounted on the wheel to accurately locate its position on the global position system (GPS) map. These sensors make the information data available to the in-vehicle data acquisition system and driver. Sensor data consists of any key features of the road such as hills, merging lanes, tunnels, or curves. The data acquisition system can analyze sensor data such as lateral acceleration, longitudinal acceleration, heading angle, yaw rate, steering angle, wheel speeds, and tire temperatures. For example, Google's self-driving hybrid vehicle [5] is a typical DA vehicle.

The driver must make the right decision at the right time in an emergency situation, in many cases, using information not easily assimilated or verifiable by cognitive processing [6]. This means is that it may not be possible for the navigation subsystem to cue the driver and wait for the proper response. The IHVMS must make the decision and execute the task providing critical time for driver override. The major emphasis is on developing a IHVMS that permits functional automation and physical integration. The objective is that the driver's cognitive and psychomotor capabilities are employed by the decision support techniques of the hybrid IHVMS design.

Vehicle Management Systems are needed now more than ever to integrate and manage: vehicle control; propulsion control; precision navigation; and vehicle instrumentation displays to provide optimal utilization of the vehicle. The functional integration levels of the current Joint Architecture for Unmanned Systems[7] (JAUS) emphasize modularity by defining strict partitioning of subsystems into software components. One of the key goals of JAUS is to promote interoperability among unmanned platforms and controllers. However, JAUS does not define the transport mechanism whereby messages are communicated between subsystems, nodes, and components. Its limitations are shortage rigid integration, decision making and a knowledge base. Decision support technology is needed to meet the expanding requirements of JAUS design.

The rapid growth and urgent need for IHVMS has resulted, however, in software for sensor control and data management being largely developed by decision support systems. Vehicle control-systems handle a wide range of different functionality ranging from safety-critical controls such as: engine control; transmission control; chassis; brake control (anti-lock brake control and anti-spin control); and to meet on board diagnosis[8] (OBD-II) requirements (warning and errors detected during run-time) driver comfort control such as: climate control; and multimedia services. The control-system consists of a number of onboard computer nodes, designated electronic control units (ECUs), which are interconnected via a controller area network[9] (CAN). Each ECU contains a number of tasks, which when executed are used to manage functionality. Since vehicle control-systems control constantly changing environments, the control-system must be a real-time system.

Most hybrid IHVMS designs emphasize the mechanical integration of vehicle and propulsion control with related utility systems. Propulsion control consists of the driver's input commands, analyzing sensor data of the gas engine and the electric motor, performing selection algorithm based on knowledge base, calculating sharing propulsions, and carrying out vehicle movement, and continuously changing power consumption during operation. Decision support IHVMS are designed to sustain a manageable driver workload while increasing total system performance, reliability, maintainability, and supportability. A primary goal of the decision support IHVMS is to enhance vehicle control effectiveness without overloading or distracting the driver. The basic approach is to categorize IHVMS functional decisions into a hierarchical control system. The top-level-decision maker is the driver. The other levels of decision making between the driver and physical subsystems are controlled by pre-programmed software subsystems. The decisions and information flowing across subsystem boundaries are collected and coordinated through the hybrid IHVMS, and then distributed to appropriate displays and effectors.

These studies show that highly automated systems should ideally result in the following conditions: system operations are simplified and managed in accordance with the driver's intent; and system functions must operate semi-autonomously with the driver's direction. The required specifications for decision support IHVMS is based upon driver-oriented and automatic-oriented design concepts. This approach uses top-down functional decomposition methods to coordinate the bottom-up physical integration.

The system operations are comprised of real-time and interactive systems. The real-time system checks system constraints, computes internal status, and identifies the ideal state for each IHVMS subsystem. The strategy for computing and evaluating is developed by decision procedures with respect to the current state of the knowledge base. The interactive subsystem runs slowly, and is involved in providing and processing information for the entire IHVMS. This provides a detailed analysis of the step-by-step decision-making process, and serves as the driver's advisor. The approach is to identify behaviors of the subsystem devices and performance requirements. The combination of algorithmic and heuristic procedures is utilized by time-constrained reasoning[10] software residing in the on-board computers of the IHVMS.

This paper presents decision support concepts that are well suited for IHVMS design, and result in required specifications for future hybrid design. In Section II, design objectives are to consolidate subsystems in one integrated system, to handle hybrid vehicle optimally and make it a cost-effective option as well. In Section III, design concepts for driver-oriented and automatic-oriented design methods are presented for setting up configurations. In Section IV, design approaches for the top-down functional decomposition and bottom up physical integration development process are discussed. In Section V, decision support techniques are outlined for the decision support IHVMS design. The Conclusion and Recommendation are presented in Section VI.

II. DESIGN OBJECTIVES

The primary objective of the hybrid IHVMS is to optimally integrate all subsystems to handle the vehicle in different environments and conditions. The secondary objective is to consolidate the functions of cruise control, propulsion control, navigation subsystem and information subsystem which would benefit from an integrated design implementation. The third objective is to reduce the cost. Usually, IHVMS are one-third the fly-away cost, and account for a large portion of the overall vehicle maintenance cost. As operation requirements become more demanding, IHVMS is getting more complex and costly.

Factors that must be considered in IHVMS design are:

- Integrity requirements from the Vehicle Control System specification guide, which are intended to incorporate driver-oriented and automatic- oriented constraints into the development of an overall design procedure to ensure operational suitability.
- A steady trend towards semi-autonomous operations, thereby increasing demands on decision support techniques; and fundamental software design, as well as development changes in promoted object-oriented design, design tool environments, and software reusability to support the intelligent behavior.

Therefore, decision support Vehicle Management Systems can be extended to encompass these new factors to enhance their desired operational capabilities in a cost-effective way.

The IHVMS concepts have evolved from experience gained with digital drive-by-wire vehicle control systems. With the incorporation of digital control systems and various subsystems it can be integrated into one system.

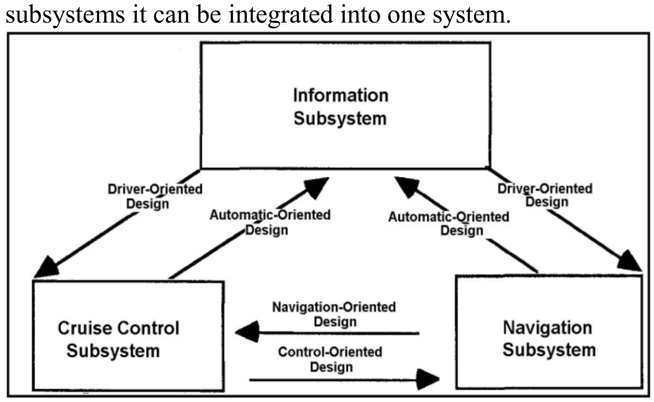

Figure 1: Components of Intelligent Hybrid Vehicle Management System

A. *Information Subsystem*

The information subsystem assists the driver with the operation of the vehicle to accomplish operation goals by notifying the driver of his/her control options and monitoring navigational plans. Figure 2 depicts the information subsystem which interacts with the navigation subsystem and the cruise control subsystem. It control media oriented center defines networking for infotainment and telematics subsystems. It records body sensor data and manages knowledge base operation between the navigation and cruise control subsystems. Data access for relating real-time data is able to be used in the vehicle cruise control subsystem. Data management can be achieved on a higher level of abstraction to provide concurrent real-time data accesses for the critical (hard real-time) and non-critical (soft real-time) data accesses to co-exist, without risking a violation of the database integrity. Data security is enhanced with respect to the highest response-times and minimized abortions.

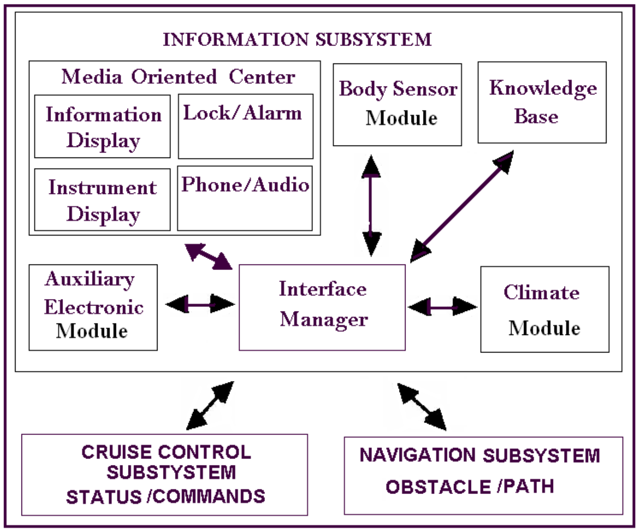

Figure 2: Information Subsystem Block Diagram

B. *Navigation Subsystem*

The navigation subsystem consists of all operation critical and non-critical functions. The navigation software executes the primary operation plan downloaded from the pre-planning subsystem. The navigation manager oversees activation, initialization, and deactivation of the subsystem. As the operation proceeds, the manager requires the cruise control manager to report if there is sufficient information and resources available (e.g., data, destination, time, fuel). The cruise control manager also provides information on the system health status: vehicle control, refueling station, terrain data, and the status at the current time. These data along with navigational planning are used to check all constraints on the allowable route and environment to generate navigational decisions. It can switch to backup plans with the driver's consent.

Figure 3 depicts the navigation subsystem which interacts with the information subsystem and coordinates with the cruise control subsystem. The modules of the subsystem consist of obstacle assessment, path planning, route planning, current vehicle state, and a navigation manager. The function of the obstacle assessment is to identify and locate vehicle threats by gaining knowledge of collision behavior, capabilities, and doctrine, and the projection of possible threats relative to its own known navigational plans. Commands are sent to the cruise control based on driver directives. The obstacle assessment derives data requirements that support the driver by making him/her aware of the obstacles. The navigation manager monitors the vehicle's current state, and determines system capabilities to decide if the cruise control can safely and reliably perform the desired operation. The information subsystem not only notifies the driver of a failure, but also provides information about the impact of the failure, as well as decision support on what recovery or reconfiguration actions to take.

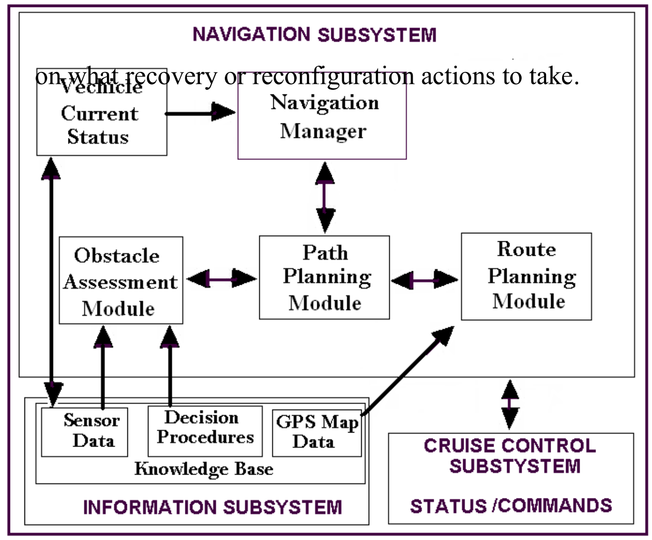

Figure 3: Navigation Subsystem Block Diagram

C. *Cruise Control Subsystem*

The cruise control subsystem is run on the drive-by-wire. It is responsible for maintaining closed loop control of the vehicle's actuators and reporting system health and monitoring safety systems. The procedures followed by the cruise control include vehicle startup, warm-up, and takeoff, the navigation plans, then parking, and shutdown. Operation procedures can be grouped into three categories: critical vehicle management, vehicle control management, and vehicle safety management which are shown in figure 4 as follows:

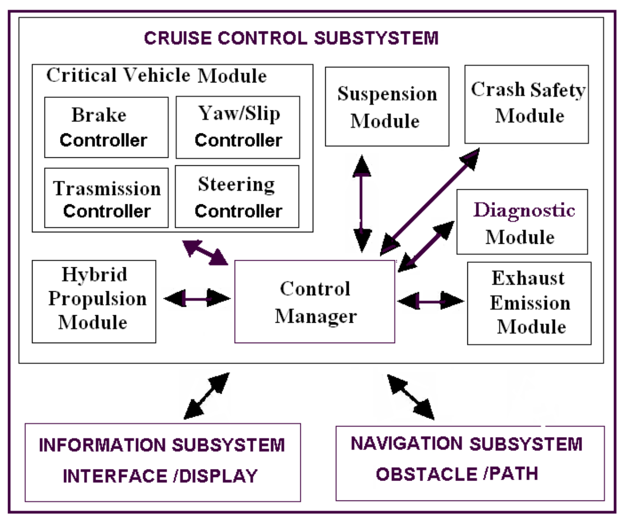

Figure 4: Cruise Control Subsystem Block Diagram

1. Critical vehicle management includes the brake controller, yaw/slip controller, transmission controller, and steering controller.
2. Vehicle control management includes the propulsion control, suspension module, and exhaust emission module.
3. Vehicle safety management includes the built-in-test

fault detection, fire detection and vehicle state data recording.

III. DESIGN CONCEPTS

To design a decision support IHVMS, the designer would have to consider integrity requirements such as structural, mechanical, electrical, and software constraints. These requirements also apply to the driver-oriented concept which allows the driver to make decisions. Similarly, the automatic-oriented concept has the automated software to do self-execution. The cruise control subsystem will reach to its maximum potential based on navigational planning support and driver proficiency.

A. Driver-Oriented Design Concept

The driver-oriented design determines different functional automation levels between the driver and IHVMS. A decision support IHVMS, called a “decision support machine” accomplishes routine, as well as highly intensive operations automatically and under driver control. For example, in a high stress situation, the driver must make the right decision at the right time to control the vehicle. The decision support IHVMS cues the driver and waits for his/her response. If the driver has reached his/her time limit and does not respond, the IHVMS must make the vehicle path control decision. At all times, the decision support IHVMS must allow driver to override. In other demanding situations, the system provides control options for the driver. Clearly, a more sophisticated vehicle will require a more automated IHVMS. This impacts the driver-oriented design requirements for the IHVMS.

Driver-oriented design is concerned with requirements such as: dashboard design; safety equipment; display equipment; and driver comfort control such as: climate control, and multimedia services. To achieve the goals of a driver-oriented design the following guidelines must be met:

1. Explicitly specify authority between the driver and the IHVMS.
2. Define performance limitations of the driver and the IHVMS.
3. Classify override criteria for vehicle-critical operations.
4. Provide sufficient real-time data information to be displayed.

The guidelines require that fundamental changes be made to support the new capabilities. These changes are not so much in hardware (controls, displays, etc.) as they are in software design and information processing. The driver’s capability to manage and operate the vehicle will evolve with successive design of the decision support system.

B. Automatic-Oriented Design Concept

The automatic-oriented design concept refers to the decision support integrity requirements for autonomous actions. Basically, hybrid IHVMS is associated with physical effects and their impact on operation performance. In one sense, there should be no function of the IHVMS actions which are not in some way related to the purpose of a particular operation. On the other hand, there are many functions which are not immediately related to operation outcomes, but interact with those that are, through physical subsystems.

An automatic-oriented design results in a functional support and functional dependency model for describing automatic integrity requirements. It is important to consider the feasibility of requirements. Feasibility is the study of impact, which occurs in the organization process during the development of a system. The impact can be either positive or negative. When the positives outweigh the negatives, then the system is considered feasible. Here, the feasibility study can be performed in three ways, which are presented as follows:

1. Algorithmic Feasibility

Algorithmic feasibility addresses the feasibility of implementing the requested performance characteristics using available software methods and algorithms. The critical question to consider is whether the characteristics that are to be implemented are a valid candidate for automation. It is often possible to utilize decision support techniques, such as heuristic processes and functional interdependencies, when there is difficulty implementing an algorithm to carry out the prescribed functional performance characteristics. An evaluation is performed to determine issues, such as complexity, sensitivity, and performance based upon the automatic integrity requirements. This step is necessary to know whether to implement the desired automation functions or to use conventional algorithms. The other factor, which indirectly includes algorithm feasibility, is the cost of the development lifecycle.

2. Computational Feasibility

Computational feasibility addresses feasibility issues such as computer architecture, data storage requirements, and instruction execution speeds. It is usually not possible to determine the parameters of computational feasibility unless the software is clearly defined. Since assessment of computational feasibility is necessary, this means that the hardware has to be considered with the algorithms and automation function development. The main concern is the adoption of a candidate automation function and the impact it will have on the data processing resources required to support its execution in the navigation subsystem.

3. Implementation Feasibility

Implementation feasibility addresses the two salient characteristics of avoidance and effectiveness. The question of implementation imposes on the cruise control subsystem concept the realization of a feasible automation function. The implementation yields three numerical results: the anticipated value of the operation, expected probability of survival, and expected probability of failure. These values are obtained from analysis of the anticipated performance of the automation functions in relation to the various portions of the functional system entities. The purpose is to combine these scores into a composite measure and evaluate their feasibility.

This evolution in the automatic-oriented design suggests inter-relationships between the low-level and high-level automations of the navigation subsystem. Its implementation will enhance the cruise control subsystem capabilities through decision support techniques. The integration of driver-centered functions is needed at the earliest stages of the design concept to meet IHVMS integrity requirements. Design approaches based upon driver-oriented and automatic-oriented concepts are described in the following section.

IV. DESIGN APPROACHES

The design approaches are two-step operations. The first step of functional analysis is top-down decomposition. The second step is system development which is the bottom-up physical integration. These steps are required to satisfy the driver-oriented and automatic-oriented requirements during the process of hybrid IHVMS development. Functional decomposition allows effective management of the design to support operation success and improve vehicle control maximizing the vehicle envelope, survivability, reliability and supportability. The top-down method generates a hierarchical structure of managed operation goals and relates them to the operations of IHVMS subsystems to achieve synergistic functional benefits. System development process requires the bottom-up integration method that results in a design that shares resources and reduces overall weight, volume, power and cost.

A. *Functional Analysis*

The functional decomposition consists of a hierarchical decomposition of operation goals. This hierarchy is, in part, dictated by operation constraints and in part by driver decision support constraints. In the analysis process, a collection of decision support alternatives are examined for each subsystem with the goal of improving subsystem operations. A suitable functional representation of the operation requires a description of both the operation structure and subsystem operations.

The functional analysis allows the implementation of subsystem control modes which benefit from the additional knowledge of the vehicle and its environment. The introduction of decision support techniques into subsystems facilitates operations necessary for functional enhancement. Functional analysis is based on a functional decomposition method that results in a better understanding of its configuration related to the capability, supportability, and maintainability characteristics.

Functional decomposition can be accomplished by functional representation and operation integrity requirements. Functional representation generates a specified IHVMS management model to perform its operation goals and test synergistic benefits of its subsystems, such as those seen in an integrated vehicle and propulsion subsystem. Operation integrity requirements describe the constraints of subsystem operations and performance.

1. Functional Representation

The functional representation provides a computer simulation model for functional analysis that relates to a generic operation model and a given benchmark scenario input. Object-oriented methods and engineering simulation are recommended to study the operation goals and identify subsystem operations. A set of subsystem operations can be thought of as instances of generic task requirements. In this case the operation consists of an object of attention, observation operations, performance criteria, and contingent actions based on input value. The operation demands temporal coordination with sortie elements, as well, as the logical constraints inherent in vehicle operational procedures.

The operational procedures are assigned and elaborated within their appropriate operation and are dependent on operation goals. The goals that are to be achieved in an operational segment are hierarchically arranged. If a goal requires that some precondition or system state be achieved (according to some criteria), before action in service of that goal is initiated, then those necessary preconditions can be expanded into subgoals. Subgoals are created by referencing to the operation definition of the higher level goal. The goal hierarchy is organized by the requirements that all subgoals contributing to a particular higher level goal must be satisfied if that high level goal is to be satisfied.

In an iterative fashion, those subgoals may require that other states are obtained in order for them to be successfully achieved. This iterative expansion of goals continues to the point at which procedures can be invoked to achieve the required subsystem operations. The process of iterative expansion generates the generic aspects of the operation goals and captures a set of subsystem operation descriptions. The arrangement of goals produces several useful properties: a detailed navigational plan; a framework for the establishment of subsystem operations; and a set of operation and vehicle constraints.

The hierarchical relationship among goals and subgoals is maintained by local information within the children of a parent goal as to their ancestors, and information within the parent goal that keeps track of its progeny. If a high level goal is to be rescheduled, the parent is removed. When the high level goal is scheduled again, the children are automatically spawned and inserted in their relative subsystem operations.

In fact, the conflict or unexpected events during plan execution reorganization of the operation plan is required to be considered. An explicit definition of the hierarchic relationship among goals and subgoals is necessary to support this reorganization. If an operation function needs to be changed, these logical dependencies are critical determinants of the degree of difficulty associated with the subsystem behavior. Functional analysis is based on a simulated operation model to generate IHVMS control and manage functional requirements by using the following steps:

a. Expand an operation into its operation dependent goals.
b. Find interconnection loops.
c. Compute steady state operating points.
d. Compute steady state input-output relations.
e. Validate subsystem operations.
f. Analyze stability, achievability and observability.

g. Compute subsystem effectiveness.
h. Compute survival functions.
i. Document the operation integrity requirements which are described in the following section.

2. Operation Integrity

Operation integrity is a key element when dealing with functional analysis. The coordination of operation integrity requirements with subsystem performance across the hierarchical structure requires the time-line descriptions of the operation. The analysis procedure, therefore, requires identification of constraint characteristics, and implementation of methods for dealing with time-constrained performance. Constraints on the overall vehicle operations result from:

a. Competitive conditional requirements of operations.
b. Survivability: fail-operational/fail-safe.
c. Vulnerability: single fail, combinations of fails, etc.
d. Performance envelope qualities of the vehicle.
e. Temporal boundaries in task execution.
f. Introduction of unexpected or uncertain conditions.

It is convenient to have a number of inferences which act on an operation and generate conditional requirements for an operation. This is accomplished by the functions: Aggregation and Generalization. The aggregation applies to a set of sub-operations of a given operation. The operation consists of the aggregated sub-operations. The generalization decomposes a given operation into a collection of sub-operations. The appropriate connections are generated from the upper level operation. These operations will have to operate on many different properties of an operation. They will create functional requirements with the appropriate properties.

B. *System Development*

System Development is the bottom-up technique for physical subsystem integration based upon functional analysis. An integrated system is one in which two or more separate subsystems need to interact in a cooperative action. Subsystem behavior allows processes to optimally utilize dedicated processors and to study operational side- effects. Subsystem integration consolidates components to use common modules, for example, subsystems are those seen in an integrated navigation system and propulsion system to share resources for improving controllability, reliability and cost. This was achieved through the elimination of under-utilized dedicated processors, and the replacement of unnecessary dedicated point-to-point wiring with digital data bus communications.

In large systems composed of many subsystems, a connection of controller, plant, measurement and gains is easily described by an optimal control system. Linear state space and transfer function behavior are well described in standard requirements on maximum or minimum gains. It is also very useful to introduce the validity range property to indicate the region of validity of the subsystem. This can be described as a set of numerical formulations.

Qualitative behavior attempts to describe system characteristics, dynamic adaptability, performance, and environment. Some of these properties are not easily done numerically. Inference mechanisms are utilized by decision support techniques to check these properties. It is therefore essential that the qualitative behavior requires procedural reasoning as well as quantitative behavior requires numerical formulation analysis.

V. DECISION SUPPORT DEVELOPMENT

Decision support techniques are implemented during software development and are based on procedural reasoning techniques and function-augmented analysis. Starting with a set of good design guidelines is essential to the software development approach. Computer software design standards will be used for this purpose. Computer software configuration comprises of top level computer software components, and lower level computer software components. These software development standards become a major portion of the developmental guidelines of a decision support IHVMS. The decision support software also includes a logic-decision tree and optimization techniques which play important roles for implementing automation features of the hybrid vehicle operational functions and driver cognitive process.

A. *Decision Support Mechanism*

Procedural Reasoning Theory provides a method of knowledge representation that specifies decision procedures and describes their effects. The fundamental assumption is that all system physical changes are caused directly or indirectly by quantitative and qualitative behavior. It induces a decision logic representation for system behavior, and allows both the deduction of what decision procedures occur in a situation, and how they might control the systems.

Decision procedures have two types of primitives that establish a framework for presenting the generic behavior of individual subsystems, qualitative constraints that present the index of performance; and quantitative states on the variables and derivatives associated with components. The procedural knowledge base determines how a subsystem works given its quantitative constraints and the knowledge of qualitative states. Processes are based upon the device's component model to derive the functioning operations. The model describes the states and constraints of the device, namely, it is a functional flow of constituent components in a system, but it does not describe the particular device or how the components are connected.

To determine the functional performances of the overall subsystem, each device's model is examined and an individual specific behavior is instantiated for it. A state-transition is used to describe the expected behavior of a subsystem in terms of states and constraints. States of the same device are mutually exclusive in the sense that a device can only operate in one of its potential states at any particular time. The behavior of subsystem operations in a particular state are characterized by a set of constraint conditions. A state transition occurs if the invocation condition defined by the state transition, of the subsystem currently operating, is satisfied. The knowledge representation includes all state-transitions for every component from its current state of operation. The knowledge representation of procedures consists of the following ingredients:

1. Device Component Model: describes a state-transition of the system.
2. System-Wide Assumptions: determine the qualitative performance.
3. Safety driving: describes the expected behavior of a subsystem in transferring threat states to safety states.
4. States: describes the potential behavior of each subsystem in a finite set of time-points representing the qualitatively distinct states and values for each quantity at each time-point.
5. Causal effects: provides the qualitative description, after the development of the state-transition.
6. Constraints: assure consistency and robustness of the process. There are five types of constraints among variables: arithmetic, functional, derivative, inequality, and conditional.

In order to predict unusual behaviors, the predictive analysis process is a technique that assesses characteristic changes (behavior events) in the behavior of the subsystem. The predictive analysis process examines the possible behaviors of a subsystem in response to a given perturbation as a sequence of time-points of abrupt-type and trend-type events. The abrupt-type event is marked by sudden changes in values occurring at a particular time. The trend-type event is marked by continuous changes in values of all variables occurring between time-points. During an abrupt-type event, instantaneous changes in the behavior of the subsystem occur. When comparing the sets of constraints before and after the state- transition, the differences are found in those constraint rules whose corresponding devices are involved in the state-transition. During a trend-type event changes in every subsystem variable is averaged in a uniform direction.

Procedural reasoning system is a framework for constructing real-time reasoning systems in dynamic environments. The architecture consists of (1) a knowledge base containing current facts about the environment; (2) a set of current destinations to be realized; (3) a set of plans, called Acts, describing how sequences of actions and tests may be performed to achieve certain goals or to react to particular situations; and (4) Intentions containing those plans that have been chosen for (eventual) execution. The requirements for decision procedures are:

1. Time-constrained response to dynamic (real-time) environment changes.
2. Assessments of current situation and data to improve system function.
3. Dynamic manipulation and modification of its knowledge base.
4. Reacts to the abrupt-type events by changing and modifying subsystem decisions.
5. Dynamic estimation, prioritization, and selection of alternatives.
6. Evaluation of information from the knowledge base and database to resolve the conflicting goals and subgoals.

As a result, all subsystems are characterized with quantitative operations under a set of qualitative constraints procedures. The inference mechanism is able to handle unpredicted events and uncertainty in the presence of incomplete knowledge.

B. Functional Augment Analysis

Functional augment analysis is a method to analyze and utilize decision support software at all levels of subsystem operations. At the top level a decision support IHVMS is a monolithic structure. It is partitioned into lower levels, each of which has its own decision support capability that is restricted by its knowledge domain. In all cases, and at all levels, the decisions may be reviewed by higher levels to insure that proper actions are taken. This generic approach to IHVMS augment analysis requires properly partitioning performance criteria into a hierarchical decision authority, and clearly defining each level's objectives and requirements.

The purpose of the augments is to convert subsystems into autonomously operation pieces. Each piece has its own goals and responsibilities, serving the needs of the next higher level of the system. Proper partitioning of IHVMS also addresses issues of physical integration. Augment analysis must consider not only the technical aspects of the implementation, but also the capabilities and business bases of the implementing organizations. Where technical issues do not mandate a particular augmentation, the physical integration of system operations should be used as a guide line. Decision support IHVMS will have a large amount of software. This is because of the integrated nature of their required flexibility and automation. The decision support software must be reliable and robust. Software libraries containing well-defined and thoroughly tested common software modules will need to be established. The design practices and packages supported by the Java language will assist in allowing a number of software packages to be made common and reusable.

These common software modules will be applied throughout the IHVMS hardware modules to improve subsystem operations and performance through decision support techniques. Vehicle control, fuel control, propulsion control, environmental control, and electrical power control functions will be constructed from hardware and software modules using the system specification. The software is complementary to the hardware and emphasizes decision support functions. Consistent with this approach, the programmable hardware modules are combined with common and decision support software modules to create a specific performance.

The steps for performing augment analysis are:

a. Use the existing functional requirements specification which the IHVMS must satisfy for proper control performance.
b. Assess new potential enhancements which will be conducted by a compatible performance study.
c. Conduct a sequence of test phases of new software along with the entire vehicle design to establish a framework.

d. Validate the augment function and its associated hardware/software elements.

The overall system architecture, as well as the knowledge based elements within this architecture, has been introduced. These elements enhance IHVMS functions and will play critical and vital roles in future vehicle design.

B. Knowledge Base Design

The main idea behind a knowledge base design is to handle information and decision procedures as an integrated whole. It is a collection of interrelated data stored with minimum redundancy to serve many users quickly and effectively. After designing an input and output, the designer must concentrate on his/her design to organize information around system requirements. The main objective is to make information access, easy, quick, inexpensive, and flexible for other users. During the design phase the following objectives are considered:

1. Data independence.
2. Retrieval decision procedures.
3. Real time updates of the operational data.
4. Controlled redundancy.
5. Accurate and integrating.

The operation executive reads command/control data messages, strips out pertinent navigational data and vehicle status data, and updates the operational databases with real time data. It will provide sufficient decision parameters to monitor the progress of the decision process. When an emergency status is detected, the obstacle assessment module executes decision procedures. It performs semi-autonomous actions either to suggest to the driver to switch to an alternate plan, or to tell the system to speed up operation to avoid newly identified obstacles and coordinate supporting elements.

Vehicle control laws can utilize the decision support techniques to decide stability margins that will have an effect on control performance and stability. The utility subsystems can be monitored by the built-in-test functions that will indicate the capability of the vehicle's safety. The knowledge base provides an intelligent way of reconfiguring subsystems, and will lead to significant improvements in the operation effectiveness.

VI. CONCLUSION AND FUTURE RESEARCH

The goal of IHVMS is to simplify driver-to-system control operations so that any command issued by the driver will cause the system to respond in the best manner possible by determining what the available resources are, and taking appropriate actions. The software development uses object-oriented and module architectures to maximize the performance, timing, reliability and safety of the system design. The design process involves the information, navigation and cruise control subsystems integration. Dependency and steps of the process suggest a typical design concept for the software system. The design concept will be implemented by user-oriented and automatic-oriented methods. The Object-Oriented Design gives the system the flexibility in future changes in the software. As the design stage approaches, functional top-down decomposition and physical bottom-up integration provide design integrity requirements, functional requirements, and physical requirements. Inference mechanism and functional augmentation provide a methodology for building an IHVMS. An open-standard commercial off-the-shelf based system is the key to building an IHVMS interoperable platform that is adaptable, flexible, scalable and available.

References

[1]. Hubert H. Chin and Ayat A. Jafari, “An Intelligent Selection Algorithm for Hybrid Vehicle Controlling.” Proceedings of the 14th International IEEE Annual Conference on Intelligent Transportation System, pp. 324-328, .2011.

[2]. CY Liang, H Peng "Optimal Adaptive Cruise Control with Guaranteed String Stability," Vehicle System Dynamics 32 (4-5): pp. 313-330, 1999.

[3]. Hubert H. Chin and Ayat A. Jafari, “Driver Associate,” Proceedings of the 36th IEEE Symposium on System Theory, pp. 400-407,.2004. http://iris.nyit.edu/~hchin

[4]. Amzajerdian, Farzin; Pierrottet, Diego F.; Petway, Larry B.; Hines, Glenn D.; Roback, Vincent E., "Lidar Systems for Precision Navigation and Safe Landing on Planetary Bodies". Langley Research Center. NTRS, May 24, 2011. Available at: http://ntrs.nasa.gov/search.jsp?R=20110012163.

[5]. John Markoff, “Google Cars Drive Themselves, in Traffic”, The New York Times, Science News, October 10, 2010.

[6]. Blanco M, Biever WJ, Gallagher JP, Dingus TA., “The Impact of Secondary Task Cognitive Processing Demand on Driving Performance,” Accid Anal Prev, 38(5), pp. 895-906, September, 2006.

[7]. http://www.jaugs.org, The JAUS Working Group, “Joint Architecture for Unmanned Systems (JAUS),” Version 3.1, Volume 2, April 2004.

[8]. http://www.obd-2.com, Homepage of the Vehicle On-Board Diagnostics II (OBD-II) , 1996.

[9]. http://www.can-cia.de, Homepage of the organization CAN in Automation (CiA) , 2004.

[10]. M. P. George and F. F. Ingrand, “Research on Procedural Resoning Systems,” Final Report, Phase 2, for NASA Ames Research Center, Moffet Field, Cal. June 1990.

Recovery from failure.