Modeling and Stochastic Analysis of Embedded Systems Emphasizing Failure Severity, Coincident Failures and Usage-Profiles

Kshamta Jerath and Frederick T. Sheldon

Software Engineering for Dependable Systems Laboratory©

School of Electrical Engineering and Computer Science

Washington State University

Pullman, WA 99164. USA.

kjerath@eecs.wsu.edu | sheldon@acm.org

Key Words - Stochastic Modeling, Reliability, Coincident Failures, Usage-Profiles.

Summary & Conclusions - The increasingly ubiquitous use of software systems has created the need of being able to depend on them more than before, and of being able to measure just how dependable they are. Knowing that the system is reliable is absolutely necessary for safety/mission-critical systems, where any kind of failure may result in an unacceptable loss of human life. This study models and analyzes the Anti-lock Braking System of a passenger vehicle. Special emphasis is laid on modeling extra-functional characteristics of severity of failures, coincident failures and usage-profiles - the goal is to develop an approach that is realistic, generic and extensible for this application domain. The strategy of modeling these characteristics (using empirical data) is innovative in terms of the approach used to integrate them into the Stochastic Petri Net and Stochastic Activity Network formalisms. The validation approach compares the results from the two separate models. The results were found to be comparable and confirm that the effect of modeling coincident failures, failure severity and usage-profiles is evident in determining overall system reliability. This work also provides a solid basis for modeling more complex systems and carrying out further supplementation and analysis.

I. INTRODUCTION

The increasingly ubiquitous use of software systems has created the need of being able to depend on them more than before; and of being able to measure just how dependable they are. Knowing that the system is reliable is absolutely necessary for safety/mission-critical systems, where any kind of failure may result in an unacceptable loss of human life (or damage to the environment). Reliability is the probability that a system will deliver its intended functionality and quality for a specified period of "time" and under specific conditions, given that the system was functioning properly at the start of this "time" period [1]. Structured models of reliability allow the reliability of a system to be derived from the reliabilities of its components. A complex embedded vehicle system (like the Anti-lock Braking System) is composed of numerous components and the probability that the system survives (efficient or acceptable degraded performance) depends directly on each of the constituent components. The reliability analysis of a vehicle system can provide an understanding about the likelihood of failures occurring in the system and an increased insight to manufacturers about inherent "weaknesses" in the system [2].

In [3], the authors presented Stochastic Petri Net (SPN) models of a vehicle dynamic driving regulation (DDR) system. Sub-system representations of the Anti-lock Braking System (ABS), the Electronic Steering Assistance (ESA), the traction control (TC) and a combined model were developed and analyzed for critical failures. In this study we model and analyze the Anti-lock Braking System of a passenger vehicle, this is but one component of the total system and there is an implicit requirement that the developed model be easily extensible and fit into a larger complex context. Special emphasis is laid on modeling extra-functional characteristics of coincident failures, severity of failures and usage-profiles.

One aspect of modeling failures occurring in the system is their severity. Severity of a failure

is the impact it has on the operation of the system. It is closely related to the threat (hazard) the problem poses, in functional terms, to the correct operation of the system [1]. Predicting the reliability/availability based on the characteristics of a model of the system provides more objective and concrete information that can be used in assessing the risk tradeoffs and integrity levels [4]. Severity is an important candidate to weight the data used in reliability calculations and must be incorporated into the model to determine the probability that the system survives, including efficient or acceptable degraded operation. Failure severity has been studied in the context of gracefully degrading systems – using Markov models to model a multiprocessor system in [5] and a portion of an air traffic control system (a set of radars) in [6].

Further, if a system does not contain any redundancy – that is, if every component must function properly for the system to work - and if component failures are statistically independent, then the system reliability is simply the product of the component reliabilities. Furthermore, the failure rate of the system is simply the sum of the failure rates of the individual components [7]. The assumption that failures occur independently (in a statistical sense) in hardware components is a widely used and often successful model for predicting the reliability of hardware devices. However, components generally interact with each other during operation, and a faulty component can affect the probability of failure of other components too [8]. Such failures are coincident in the sense that failure of one component increases the probability of failure of another.

Several researchers have considered the problem of modeling correlation between failures. Two schools of thought emerged, differentiated by the definition of the basic events of interest, the two approaches are called Correlated failures and Differentiated causes [9]. The correlated failures approach was first considered by Eckhardt and Lee [10], and their work was later

extended by Littlewood and Miller [11]. The differentiated causes approach was first proposed by Arlat, Kanoun and Laprie [12] and later adopted by others.

The reliability of a system also depends on its usage profile – users interact with the system in an intermittent fashion, resulting in operational workload profiles that alternate between periods of “Active” and “Passive” use. Reliability is concerned with the service that is actually delivered by the system as opposed to a system’s capacity to deliver such service [13]. Specifically, while considering usage profiles, faults need not necessarily cause failures since they can be repaired; failures occurring during “active” use of the system only should contribute to reliability calculations. Some investigations in this field have been mainly experimental, using empirical data from measurements of real systems to correlate workload with various measures of dependability [14, 15]. On the analytic side, probabilistic models have been used to obtain workload-related dependability measures [5, 16-18].

This research contributes by incorporating the characteristics of failure severity, coincident failures and usage-profiles into the model developed for the Anti-lock Braking System of a passenger vehicle [19]. These characteristics have never been modeled together for this system, generating a potentially more realistic model (with real data being used to model failure rates). The modeling strategy adopted in this study is innovative in terms of the approach used to integrate the above characteristics into the Stochastic Petri Net (SPN) and Stochastic Activity Network (SAN) formalisms.

SPNs and SANs provide concise representations of the system being modeled and can be used to automatically generate the underlying Markov models. The modeling approach needs to overcome the two most common challenges in modeling using Markov models – large state space and stiffness. Further, modeling and experiments on the real system (measurements) are

two totally different but complementary approaches in performance evaluation methodology. Since, it is beyond the scope (and the means) of this research to validate the results from the analytic experiments against real data, two different stochastic formalisms have been used to carry out the reliability analysis (see Fig. 13). The results, from the analyses of each of the (SPN and SAN) models developed, using the Stochastic Petri Net Package [20] and the UltraSAN [21] tools respectively, are discussed and compared.

II. THE ANTI-LOCK BRAKING SYSTEM DESCRIPTION

The Anti-lock Braking System (ABS) is an integrated part of the total braking system in a vehicle. Applying excessive pressure on the brake pedal, or panic slamming the brake pedal, can cause wheels to lock up and possibly send the vehicle careening into a terrifying skid. Excessive brake pedal pressure often occurs in an emergency or adverse situation, such as wet or icy roads. The ABS prevents wheel lockup during an emergency stop by modulating the brake pressure and permits the driver to maintain steering control while braking.

A. Components and Functioning of the ABS

The ABS consists of the following major components. (1) Wheel Speed Sensors - These measure wheel-speed and transmit information to an electronic control unit, (2) Electronic Control Unit (Controller) - This receives information from the sensors, determines when a wheel is about to lock up and controls the hydraulic control unit, (3) Hydraulic Control Unit (Hydraulic Pump) - This controls the pressure in the brake lines of the vehicle, and (4) Valves - Valves are present in the brake line of each brake and are controlled by the hydraulic control unit to regulate the pressure in the brake lines.

Fig. 1 displays the top-level schematic of the system showing the interconnections between the components. When a driver applies brakes on an ABS-equipped vehicle, wheel sensors monitor

the rotational speed of each wheel. The electronic control unit (ECU) "reads" signals from the sensors and compares the speed of each wheel. If one wheel is slowing at a faster rate than the others, the ECU sees that the wheel is beginning to lock up. The ECU then orders the hydraulic control unit (HCU) to reduce the line pressure to that wheel's brakes. The HCU reduces the pressure in that particular brake line by controlling the valves present there. Once the wheel resumes normal operation, the controls restore pressure to its brake. Depending on the system, this "pulsing" of brake line pressure can occur at up to 15 times per second. The result is that the tire slows at the same rate as the vehicle, with the brakes keeping the tires very near the point at which they will start to lock up. This gives the system the highest steering capability.

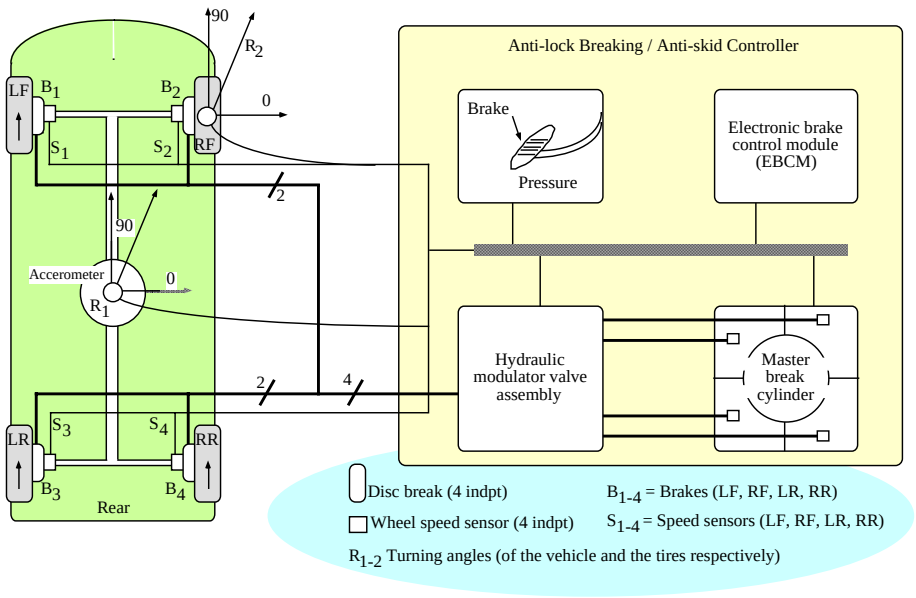

Fig. 1: Top-level schematic showing sensors, processing and actuators

Since the system we study here is very complex, this prevents us from making a direct analysis. A series of abstraction steps are needed to obtain system measures from the real system. Initially the system model is created at an abstract level and the data collected from system measurements are used to parameterize the abstract model. In the second abstraction step the computational model is created which allows an easier and more efficient system analysis [22]. The key element therefore in our modeling approach was to identify the essential components of

the system, the different ways in which they interact and introduce various assumptions. Table I provides a list of the parts that are considered in this analysis, along with the respective failure rates[1] associated with critical failure states.

Table I: Component failure rates associated with critical failure states

Component	#	Base Failure Rate	Probability		
			Degraded operation	Loss of Stability	Loss of Vehicle
Wheel Speed Sensor	4	2.00E-11	0.38	0.62	-
Pressure Sensor	4	1.50E-11	0.64	0.36	-
Main Brake Cylinder	1	1.00E-11	-	-	1.0
Pressure Limiting Valve	2	6.00E-13	-	0.22	0.78
Inlet Valve	4	6.00E-13	-	0.18	0.82
Drain Valve	4	6.00E-13	-	0.19	0.81
Toggle Switching Valve	2	6.00E-13	1.0	-	-
Hydraulic Pump	2	6.80E-11	-	-	1.0
Pressure Tank	2	2.00E-12	-	-	1.0
Controller	1	6.00E-12	0.2	0.4	0.4
Tubing	1	3.00E-12	0.33	-	0.67
Piping	1	4.00E-12	0.33	-	0.67

B. System Assumptions

The various assumptions made about the system are:

1) Three modes of operation: For the purpose of this discussion, the different modes of operation of the system (in presence/absence of failures of different severity) are assumed to be: (i) *normal operation*, (ii) *degraded operation*, and (iii) *lost stability mode*; in increasing order of severity. Critical failures seriously impact the operation of the system, and are assumed to cause *loss of vehicle*. Further, if sufficient components of the system have failed to impact the system operation (either degraded operation or lost stability mode), the sum of those failures is assumed to be critical, causing loss of vehicle.

[1] The data was obtained from DaimlerChysler. The failure rates listed in Table 1 however are dummy values. The real values we had are protected under a non-disclosure agreement.

2) Lifetime of passenger vehicle: Essentially the average hours of operation for a passenger vehicle range from 300-600 hrs /year and the average lifetime is 10-15 years. Thus, the average life span of a passenger vehicle ranges from 3000 – 9000 hrs. This estimate is important while considering the duration for which to carry out the reliability analysis.

3) *Interdependencies between components:* To model coincident failures, several dependencies among system components are assumed. Only those inter-relationships between components depicted as solid arrows in Fig. 2 are explicitly modeled in the stochastic models. All other possible inter-relationships between components (only some of them depicted as dashed arrows in the figure) have been ignored. Further, for modeling purposes, we assume a four channel four sensor ABS [23]. The model can be easily modified to represent other ABS schemes.

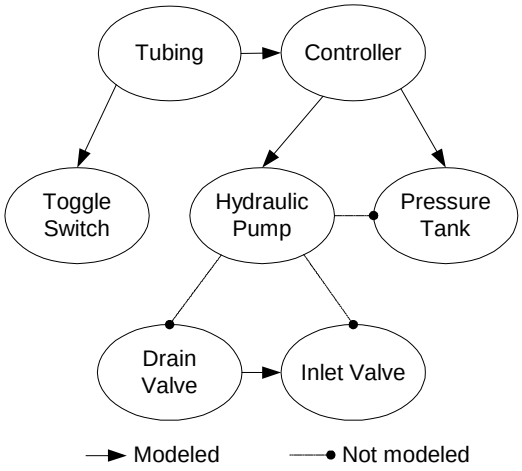

Fig. 2: Interdependencies between components

III. STOCHASTIC PETRI NET MODELS

The SPN models developed to model severity of failures and coincident failures for the ABS, as well as the SPN models developed to model usage-profiles for the ABS are presented here [19]. The SPNs were input to the Stochastic Petri Net Package (SPNP v. 6) tool in CSPL (C-based Stochastic Petri net Language). Here, the models are discussed in Petri net form for clarity. Code is presented for explanation wherever necessary.

The ABS is represented as a combination of all the important components it consists of, as shown in Fig. 3. The components are sorted into two groups: *central* and *axle*. The components

under *axle* are further segregated according to the corresponding *wheel* – FRWheel (Front Right Wheel), FLWheel (Front Left Wheel), RRWheel (Rear Right Wheel) and RLWheel (Rear Left Wheel). This division into groups is representative of the number of a given component present in the system. A component like the Wheel Speed Sensor, one for each wheel, finds its place under each of the four wheel categories (FRWheel, FLWheel, RRWheel and RLWheel). A component like the Hydraulic Pump, one for each axle, finds its place in the *axleCentral* group under the *axle* place. A component like the Main Brake Cylinder, of which there is only one instance, finds its place under the *central* category. Each component has its own model, shown as dashed rectangles in Fig. 3.

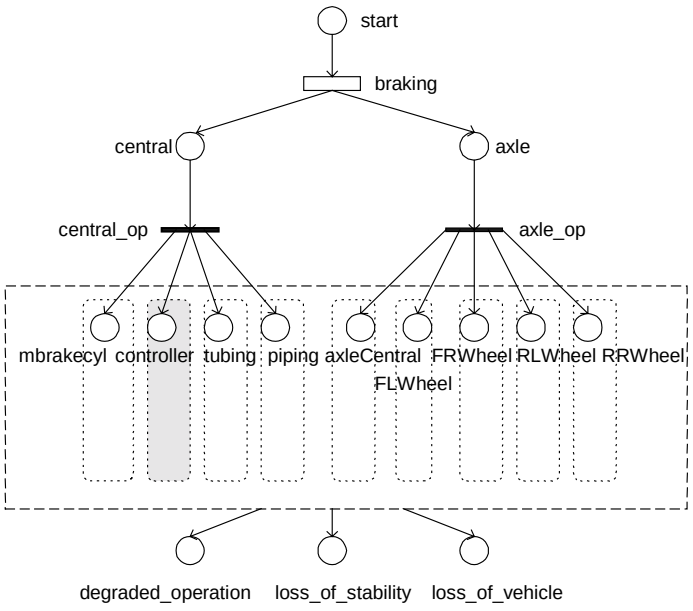

Fig. 3: SPN model for the ABS

A. Modeling Severity of Failures

The model shown in Fig. 3 also depicts the operation of the ABS under normal, degraded and lost stability conditions. The places *degraded_operation*, *loss_of_stability* and *loss_of_vehicle* model the severity of failure. The system is functioning normally when there are no tokens in any of these three places. The model is initialized with a single token in the *start* place. When the *central_op* and the *axle_op* transitions fire, a token is deposited in each place that represents a component of the ABS. The operation of each component is now independent of every other component (except where coincident failures are modeled explicitly as explained in the next

section).

The model of a component of the ABS is shown in Fig. 4. The component depicted here is the controller. Every component either functions "normally" as shown by the *controllerOp* transition or "fails" as shown by the *controllerFail* transition. To allow a Markov chain analysis, the time to failure of all components is assumed to have an exponential distribution. A failed component may either cause degraded operation, loss of stability or loss of vehicle (as represented by the *controllerDegradedOp*, *controllerLOSOp* and *controllerLOV* immediate transitions respectively). The probability of any one of these three transitions occurring (obtained from measures on the real system) is different for each component and is shown in Table I. When the failure causes either degraded operation or loss of stability, the component continues to operate (token recycled back to the *controller* place), though the failure rate increases by two and four orders of magnitude respectively. Loss of vehicle (indicated by a token in *loss_of_vehicle*), extreme loss of stability (indicated by three tokens in *loss_of_stability*) or extreme degraded operation (indicated by five tokens in *degraded_operation*) signify critical failures and determine the halting condition for the model.

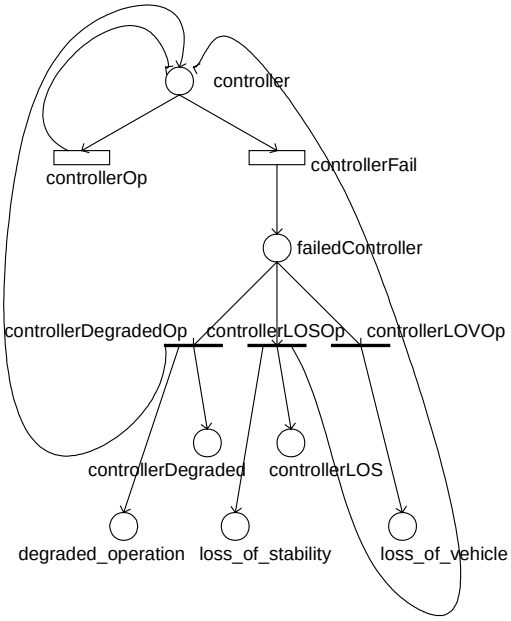

Fig. 4: SPN model of the controller

B. Modeling Coincident Failures

To model coincident failures, several dependencies among system components are assumed, as shown in Fig. 2. Coincident failures are modeled in a manner similar to severity of failures.

"Coincidence" of failures of two components is modeled by causing the failure of one component (to degraded operation or loss of stability) to increase the failure rate of the dependent component. The failure of a component A to a degraded mode causes the failure rate of a "related" component B to increase by two orders of magnitude. The failure of component A to a lost stability mode causes the failure rate of a "related" component B to increase by four orders of magnitude. (There was no data available to confirm or validate this assumption for modeling coincident failures.)

The function that calculates the failure rate of the transition *controllerFail* is shown in Fig. 5. It is assumed that a tubing malfunction affects the operation of the controller. Hence, while calculating the failure rate of the controller, the normal rate is increased by two orders of magnitude if the tubing has failed causing degraded operation (indicated by a token in the *tubingDegraded* place). Only a few coincident failures have been represented in the model. However, coincident failures between other components (or among more than two components) can be easily modeled by suitably modifying the failure rate function (as in Fig. 5) of the relevant components using the same rule.

```
double controllerRate()
{
    double controller_rate = 0.0000006;

    if (mark("controllerLOS") > 0) return controller_rate * 10000;
    if ((mark("controllerDegraded") > 0) || (mark("tubingDegraded") > 0))
        return controller_rate * 100;
    return controller_rate;
}
```

Fig. 5: Variable failure rate function to model coincident failures

C. Modeling Usage-Profiles

The global model of the ABS, represented as a combination of all the important constituent components, remains unchanged (Fig. 3). To incorporate the usage-profiles in the ABS model, the model of each individual component, like the *controller* depicted in Fig. 4, is extended as shown in Fig. 6. The figure shows the *controller*, with the bold lines indicating the additions to

the model. In case of a failure (*failedController*), the model differentiates between the two situations regarding whether the system was in active use (along the branch to transition labeled *mu*) or not (along the branch to transition labeled *alpha*). The parameter 1/*mu* indicates the mean duration of active use while the parameter 1/*alpha* indicates the mean duration of passive use. The rate of active use is assumed to be exponentially distributed.

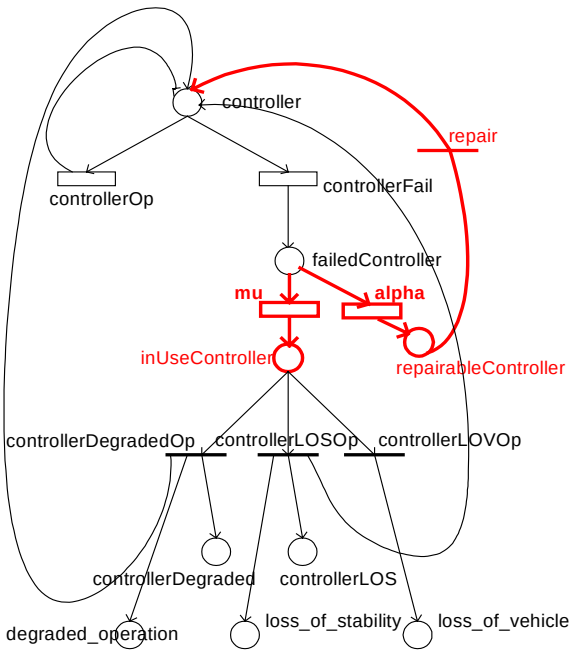

Fig. 6: SPN model of controller with usage-parameters

In the case where the failure occurs during the active period (*inUseController*), the system either continues to operate in the degraded (*controllerDegradedOp*) or lost stability mode (*controllerLOSOp*), or causes loss of vehicle (*controllerLOVOp*). In the case where the failure occurs during passive use of the system (*repairableController*), the fault can be repaired and an infinite repair rate is assumed (all repairs occur instantaneously). The system continues to operate as if no failure had occurred. The model can be extended to associate a cost with each time the failed component must be repaired, if required.

To work around the state explosion problem occurring due the evident increase in the number of states in the model as shown in Fig. 6, the model is

```
double controllerRate()
{
        double controller_rate = 0.0000006;

        // usage parameter
        controller_rate += controller_rate * mu();

        if (mark("controllerLOS") > 0) return controller_rate * 10000;
        if ((mark("controllerDegraded") > 0) || (mark("tubingDegraded") > 0))
                return controller_rate * 100;
        return controller_rate;
}
```

Fig. 7: Variable failure rate function to model usage-parameters

simplified to incorporate the usage parameters while calculating the failure rate itself for each component. The modified function for calculating the failure rate in light of the usage-profile is shown in Fig. 7. Essentially, the failure rate (considering only usage-parameters) is the sum of the failure rates *mu* and *lambda*. The value of these usage parameters was factored by the actual failure rate of the component to avoid stiffness in the model. The value of *mu* is assumed to be 2.5 for infrequent use periods (low-usage) and 250 for frequent active use periods (high-usage).

D. Extensibility of the SPN Models

The SPN models developed for modeling coincident failures and severity and usage-profiles are easily extensible. The global SPN Model can be extended to include other components deemed relevant to the ABS by including their corresponding sub-models. The sub-models, in turn, would be simple reproductions of the sub-models for other components with different failure rates and probabilities. The model, developed for the four channel four sensor ABS, can be adapted to model other schemes of the ABS, by suitably changing the numbers of the relevant components modeled (by either removing/adding the respective place, or updating the failure rate). Inter-dependencies between other components (or among more than two components) culminating in coincident-failures can be modeled by updating the failure rates of the relevant components.

Different categorizations for severity of failure can be used by simply updating the sub-models of the components to include the necessary places depicting the severity level. For example, the model representing the controller component (Fig. 4) can be updated to have a transition (in addition to *controllerDegradedOp*, *controllerLOSOp* and *controllerLOVOp*) and a place (in addition to *degraded_operation*, *LOS* and *LOV*) to represent an additional failure severity level. The SPN model representing usage profiles can be updated to represent different usage-

parameters or intensity of workload by simply changing the value of *mu*. The model can also be extended to associate a cost with each time the failed component must be repaired, by adding an additional place to keep track of the number of times the component has been repaired denoted by the number of tokens in this place.

IV. STOCHASTIC ACTIVITY NETWORK MODELS

The SAN models developed to assess severity of failures and coincident failures for the ABS, as well as the SAN models developed to model usage-profiles for the ABS are presented here [19]. The SANs were input to the UltraSAN (v. 3.5) tool graphically. Here, the composed model and the individual subnet *Central_2* are discussed. Code is presented for clarity, wherever necessary.

The composed model for the ABS is shown in Fig. 8. The model consists of three individual SAN sub models: *Central_1*, *Central_2* and *Wheel*. The *Wheel* subnet is replicated four times to model the four wheels of the vehicle. The division into these three sub-categories is done to facilitate the representation of coincident failures. As depicted in Fig. 2, the inlet valve and the drain valve (in the *Wheel* subnet) are correlated, and so are the components listed under the group *Central_2* (gray-colored subnet). All components under *Central_1* are assumed to be independent of each other (for the purpose of this study). Such a distribution/categorization avoids replicating of subnets where unnecessary (for modeling severity and coincident failures) and thereby mitigates the state explosion problem.

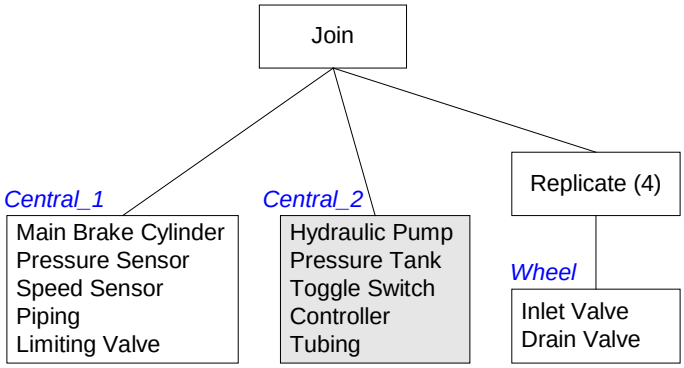

Fig. 8: The composed model of the ABS

A. *Modeling Severity of Failures*

All subnets when combined to form the composed model share some common places: *degraded*, *LOS*, *LOV* and *halted*. The first three places model the severity of failure, while the *halted* place is used to determine the halting condition. The *Central_2* subnet is shown in Fig. 9. The presence of tokens in *degraded*, *LOS* and *LOV* represent degraded system operation, loss of stability and loss of vehicle respectively (the same concept used in the SPN models). The system is operating normally when there are no tokens in any of these three places.

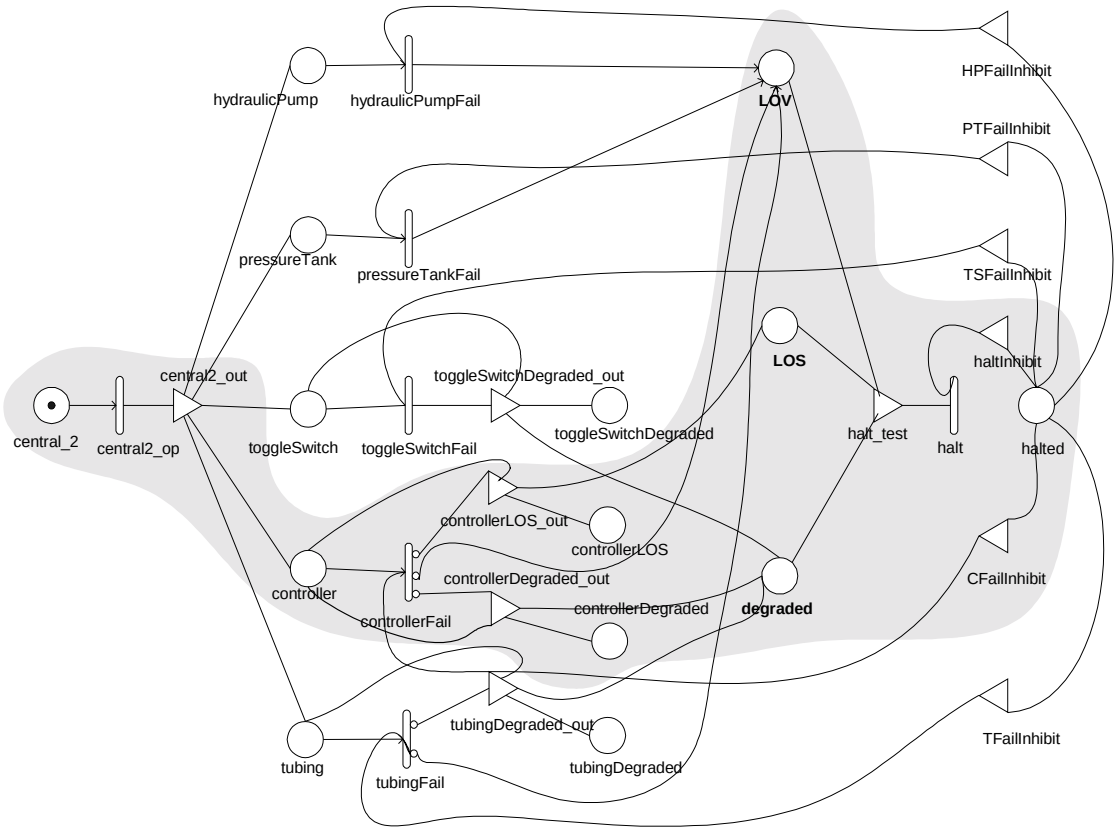

Fig. 9: *Central_2* subnet with the controller component highlighted

The subnet is instantiated with a single token in the *central_2* place. The *central2_op* activity fires and deposits a token in each of the five places: *hydraulicPump*, *pressureTank*, *toggleSwitch*, *controller* and *tubing*. The portion of the subnet for the *controller* component is highlighted in

Fig. 9 and discussed here in the context of severity of failures. The *controllerFail* activity models the failure of the controller. There are three possible outcomes of this activity. The *controller* either fails causing degraded operation (with probability 0.2, output gate *controllerDegraded_out*), or causes loss of stability (with probability 0.4, output gate *controllerLOS_out*), or causes loss of vehicle (with probability 0.4, output to *LOV*). In the former two cases the controller continues to operate in a degraded manner, as is evident by the recycling back of the token to the *controller* place. Further, the failure rate in this situation increases by two (for degraded) and four (for loss of stability) orders of magnitude respectively. The code snippet that achieves this is shown in Table II.

Table II: Activity rates model severity and coincident failures

Activity	Rate	Probability		
		Case 1	Case 2	Case 3
controllerFail	MARK(controllerLOS)!=0? controllerRate*10000: (MARK(controllerDegraded)!=0 ‖ MARK(tubingDegraded)!=0 ?controllerRate*100 :controllerRate)	0.4	0.4	0.2
hydraulicPump Fail	MARK(controllerLOS)!=0? hydraulicPumpRate*10000: (MARK(controllerDegraded)!=0 ? hydraulicPumpRate*100 : hydraulicPumpRate)	1.0	-	-

UltraSAN requires the failure rate to be specified in a single statement, hence the use of the special if-then-else construct available in the C programming language. Consider the *controllerFail* activity. If the controller fails causing degraded operation (i.e., MARK(controllerDegraded)!=0), it continues to function manifest by recycling the token back to the *controller* place, and the failure rate for the *controllerFail* activity increases by two orders of magnitude (i.e. controllerRate*100). Similarly, if the controller fails causing loss of stability (i.e., MARK(controllerLOS)!=0), (again) it continues to function manifest by recycling the token back to the *controller* place, and the failure rate for the *controllerFail* activity increases by four orders of magnitude (i.e. controllerRate*10000).

B. Modeling Coincident Failures

As in the SPN models, “coincidence” of failures of two components is modeled by causing the failure of one component (to degraded operation or loss of stability) to increase the failure rate of the dependent component. The failure of a component A to a degraded mode causes the failure rate of a “related” component B to increase by two orders of magnitude. The failure of component A to a lost stability mode causes the failure rate of a “related” component B to increase by four orders of magnitude. Table II shows the rates for the activities modeling the failure of the controller and the hydraulic pump (other component failure rates as modeled in a similar manner).

Since UltraSAN requires the failure rate to be specified in a single statement, the special if-then-else construct available in the C programming language is used. Consider the *controllerFail* activity. Since a failed tubing (in degraded mode) is assumed to affect the failure rate of the controller, if the number of tokens in the *tubingDegraded* place is not zero (i.e., MARK(tubingDegraded)!=0), the failure rate for the controller increases by two orders of magnitude (i.e., controllerRate*100). Similarly, for the *hydraulicPumpFail* activity, we have assumed that a failed controller affects the failure rate of the hydraulic pump. Thus, the failure rate for the hydraulic pump increases by four orders of magnitude if the controller has failed causing loss of stability. Also, the failure rate for the hydraulic pump increases by two orders of magnitude if the controller is operating in a degraded mode. Only a few coincident errors have been modeled. However, coincident failures between other components (or among more than two components) can be modeled in a similar fashion.

C. Modeling Usage-Profiles

The composed SAN model for the ABS remains unchanged while modeling usage-profiles.

The individual component's model within each subnet needs to be updated to handle usage, as done in the SPN models.

To avoid the state explosion problem occurring due the increase in the number of states in the model (for a component in passive and active modes), the model is simplified to incorporate the usage parameters while calculating the failure rate itself for each component. The modified construct calculating the rate for each failure activity in light of the usage-profile is shown in Fig. 10. The translation to the simple C if-else statement is also presented for clarity.

```
MARK(componentBLOS)!=0 ?
(componentARate+componentARate*mu)*10000 :
(MARK(componentBDegraded)!=0 ?
(componentARate+componentARate*mu)*100 :
(componentARate+componentARate*mu)).
```

Is Equivalent to:

```
if(MARK(componentBLOS)!=0)
   return (componentARate+componentARate*mu)*10000;
else if(MARK(componentBDegraded)!=0)
   return (componentARate+componentARate*mu)*100;
else        return (componentARate+componentARate*mu);
```

Fig. 10: Construct to model usage-profiles

The parameter 1/*mu* indicates the mean duration of active use of a given component. To calculate the failure rate of the component, the actual failure rate is added to the active usage rate (*mu* factored by the actual failure rate to avoid stiffness due to the evident difference in the orders of magnitude). The remaining constructs for severity and coincident failures remain unchanged. The value of *mu* is assumed to be 2.5 for infrequent active use periods (low-usage) and 25 for frequent active use periods (high-usage).

D. Extensibility of the SAN models

The SAN models developed for modeling coincident failures and severity, and usage-profiles are easily extensible. The composed SAN model can be extended to include other components deemed relevant to the ABS by adding other subnets, or including the component(s) as part of an existing subnet. A new component is modeled in the same way as other existing components, with its own failure activity and the corresponding output cases and probabilities.

Adding more components into the model can however aggravate the state space explosion problem due to the interleaving of the different token configurations within the subnet as well as among all the subnets in the composed model. The strategy should avoid using multiple places to denote multiple instances of the same component where possible e.g. instead of using two places to denote two axles, use a single place with the associated activity having a failure rate twice the failure rate for one axle. If it is absolutely imperative to model the two axles separately, then use Replicate, because then UltraSAN can take advantage of the State Lumping Theorem that allows the generation of a smaller state space.

The model, developed for the four channel four sensor ABS, can be adapted to model other schemes of the ABS, by suitably changing the numbers of all the relevant components modeled (by either removing/adding the respective place, or updating the failure rate as described above). Different categorizations for severity of failure can be realized by simply modifying the component sub-models to include the necessary places representing the severity level. The levels of severity can also be altered by changing the number of tokens in each of the “severity” places necessary to cause the system to halt (i.e., the halting condition). Different levels of severity can also be modeled by multiplying the failure rate of the affected component by a different scalar (other than 100 and 10000 for degraded mode and lost stability respectively). Inter-dependencies between other components (or among more than two components), which cause coincident-failures, can be modeled by updating the rates of the activities that model failure of those components.

The SAN model representing usage profiles can be updated to represent different usage-parameters or intensity of workload by simply changing the value of *mu* in Fig. 10. The model can be extended to associate a cost with each time the failed component must be repaired; by

adding an additional place to keep track of the number of times the component has been repaired (denoted by the number of tokens in this place).

V. RELIABILITY ANALYSIS RESULTS AND COMPARISON

The reliability of the system at time *t* is computed as the expected instantaneous reward rate at time *t*. To determine the reliability of the system, transient analysis of the developed models was carried out and the reliability measured between 0 and 50K hrs. The time duration was deliberately conservative, even though the average life span of a passenger vehicle ranges from 3000 – 9000 hrs, the reliability measures were determined for up to 50K hrs. The transient analysis was carried out using the Stochastic Petri Net Package (SPNP) version 6 (for the developed SPN models) and UltraSAN version 3.5 (for the developed SAN models) on a Sun Ultra 10 (400 MHz) with 500MB memory.

Since, it is beyond the scope (and the means) of this research to validate the results from the analytic experiments against real data, two different stochastic formalisms have been used to carry out the reliability analysis. The transient analysis of the developed SPN models resulted in 164,209 tangible markings, of which 91,880 were absorbing. The approximate running time of the solver on the models was 144-168 hrs. The developed SAN models were solved at a tolerance of 1e-09, and resulted in the generation of 859,958 states. The approximate running time of the solver on the models was 120-144 hrs. The results, from the analyses of each of the (SPN and SAN) models developed, using the SPNP and the UltraSAN tools respectively, are presented and compared in this section.

A. Results for Models Representing Coincident Failures and Severity of Failure

The results for both SPN and SAN models representing coincident failures and failure severity are shown in Fig. 11. The Y-axis gives the measure of interest - the reliability; while the time

range (0 to 50K hrs) is shown along the X-axis. The interval between the points did not remain constant along the entire time range and therefore the X-axis is not linear and should be considered (i.e., the interval size increases with time) when viewing the results graphs. As expected, the reliability steadily decreases with time. The box highlights the range of average lifetime of a vehicle.

The curves for the two SPN models for coincident failures (the model representing severity and coincident failures and the model not representing them) are completely overlapped. The Mean Time to Failure (MTTF) for the model with coincident failures (784,856 hrs) is approximately 421 hrs less than the model without coincident failures (785,277 hrs). For the *limited*[2] number of coincident failures that have been modeled, the difference of 421 hrs in the two cases is considered well within the confidence interval.

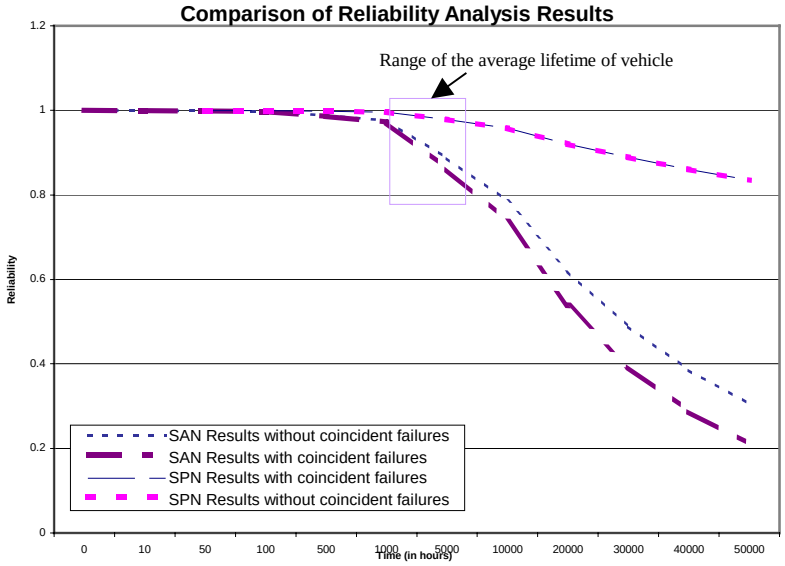

Fig. 11: Comparison of SPN and SAN results for the models representing severity and coincident failures

For the SAN models, the reliability functions diverge perceptibly after around 1K hrs of operation, and the difference continues to increase with time. At 50K hrs, the reliability has dropped down to 0.21 when coincident failures are modeled, and down to 0.30 when coincident

[2] One may speculate that there is some kind of relationship (perhaps linear) between the number of dependencies modeled and the difference observed in the graphs and the MTTF values.

failures are not modeled, a difference of 0.09 in reliability in the two cases within 50K hrs. The MTTF at 50K hrs when coincident failures are not considered is 29,167 hrs, and when considered is 25,409 hrs, a difference of 3,758 hrs.

The difference in the range of reliability values produced by the two different formalisms may be attributed to the way the reliability reward is defined in each. The SPN reward rate was defined as a single set of discrete 0/1 values, while the SAN reward rate function models a range between 0 and 1 (a function of the number of tokens in the *degraded*, *LOS* and *LOV* places). Therefore, the different rewards accumulate at different rates, and this explains the disparity in the reliability values at any given point in time.

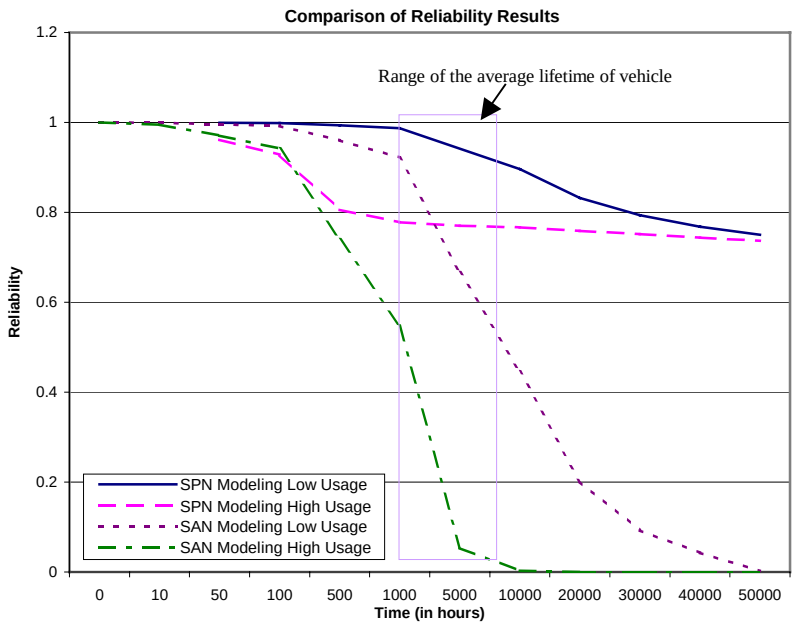

Fig. 12: Comparison of SPN and SAN results for the models representing usage-profiles

It is evident that representing severity and coincident failures in the model contributes to system reliability prediction. Such a prediction may be closer to how the real system will behave under the assumptions that have been made.

B. Results for Models Representing Usage-Profiles

The results for both SPN and SAN models representing coincident failures and failure severity are shown in Fig. 12. For the SPN models, the reliability of the system with heavy usage decreases alarmingly within the first 1K hrs of operation, while the reliability of the system with

moderate usage decreases perceptibly only after 2.5K hrs of operation and then steadily afterwards. Also, the MTTF for the high usage case is approximately 771,023 hrs as opposed to 775,112 hrs for the low usage case, a difference of 4,089 hrs.

For the SAN models, the reliability of the system with heavy usage starts decreasing alarmingly after the first 100 hrs of operation, while the reliability of the lightly used system decreases only perceptibly after 100 hrs of operation and then steadily afterwards. Considering the expected lifetime period of the vehicle (3,000-9,000 hrs), the reliability for the high-usage profile drops from around 0.55 down to approximately 0.05. For the same duration, the reliability for the low usage drops from 0.9 to only 0.5, a difference of approximately 0.45 after 10K hrs of operation. Thus, aggressive use of the system, causes the reliability to drop more rapidly (as expected) than when the system is used conservatively.

Table III: Comparison of criteria and results for SPN and SAN models

Criteria/Results	SPN Model	SAN Model
Assumptions	Same	Same
Reliability Measure	Defined as a set of 0/1 rewards	Defined as a function of the number of tokens in degraded, LOS and LOV places
Number of states generated	164,209 states	859,958 states
Environment/Platform	Same	Same
Running Time of Solvers	144-168 hrs	120-144 hrs
Reliability at 9K hrs (With severity and coincident failures vs. without)	0.95792578 vs. 0.95792653	0.73672 vs. 0.786
Difference in reliability at 9K hrs (Without severity & coincident failures minus with)	0.00000075	0.04928
Reliability at 9K hrs (Low Usage vs. High Usage)	0.89621556 vs. 0.76658329	0.4455167 vs. 0.3130521
Difference in reliability at 9K hrs (Low Usage minus High Usage)	0.12963227	0.1324646

Table III provides a tabularized summary listing the comparison criteria and results from both the SPN and SAN models. As stated before, the assumptions and the environment/platform used for the analysis were the same in both cases, while the reliability reward measure was different.

The number of states generated for the SPN models was much lower than the number for the SAN models. However, the running time of the solvers was longer for the SPNs. The reliability results at 9K hrs (the expected end-of-life time point) are also provided.

VI. CONCLUSIONS AND FUTURE WORK

The objective of this research was to develop a generic (including extensible and realistic) framework for analyzing an embedded vehicle system focusing on severity of failures, coincident failures and usage-profiles using SPNs and SANs. This objective was achieved by representing these characteristics in the special case of an Anti-lock Braking System of a passenger vehicle. Modeling challenges of state explosion and stiffness were overcome by limiting the state space size. This may be bore out by increasing the state space that makes the models intractable. The basis of creating the models was an industrial-strength system characterized by empirical data. The modeling strategy was explained and the extensibility of the models developed was discussed. Two different stochastic formalisms – Stochastic Petri Nets and Stochastic Activity Networks, were used to analyze the developed models for reliability measures. The results from both formalisms were compared and discussed, in the absence of other validation procedures (see Fig. 13). The goal of developing an approach that is generic and extensible for this application domain was achieved by discussing the extensibility of the developed models to incorporate greater complexity and and/or modifications with respect to the aforementioned assumptions.

A. Conclusion

The characteristics of failure severity, coincident failures and usage-profiles were successfully incorporated into the model developed for the Anti-lock Braking System of a passenger vehicle. This resulted in generating a potentially more realistic model (with real data being used to

determine failure rates). This study established the degree of complexity and the level of abstraction that is feasible to model and solve utilizing the available resources. Comparison of the models developed using the two separate modeling formalisms, SPNs and SANs, indicate that there is no clear winner between the two formalisms. Even though SANs provide more expressive power and compactness to the models developed, the details that are clearly visible in the SPN models are hidden in the SAN models, making some details less intuitive.

The results from the analyses of the models developed indicate the divergence of reliability measures in the cases when the extra-functional characteristics of severity and coincident failures were and were not modeled. The difference in reliability measures for markedly different system workloads was also identified. Because the model is an abstraction of a real world problem, predictions based on the model must be validated against actual measurements observed from the real phenomena. Using two different stochastic formalisms for analysis and comparing the results did not compensate for the absence of validation procedures for analysis results. Yet, the results justified the modeling strategy adopted. They also highlighted the importance of modeling severity, coincident failures and usage-profiles while examining system reliability. Suitable validation procedures can provide helpful feedback for refining the model and making it even more realistic. Analysis of such a realistic model provides the basis for verifying that the requirements for system safety have been satisfied. These results can also help to determine prognostics (predict when to schedule maintenance).

This research has successfully established a framework for investigating system reliability. The contribution of this research to the automotive industry offers greater insight into the strategy for developing realistic models. The information from the results is also useful in making design decisions. Results provide greater insight to manufacturers about “weak links” in the system, and

increased understanding of which components need to be highly reliable to potentially increase overall system reliability. This study can also be the basis of numerous other studies, building up on the framework and investigating other areas of interest, as presented in the next section.

B. Future Work

The Anti-lock Braking system is a small part of the DDR (Dynamic Driving Regulation) system, which consists of other subsystems like the Electronic Steering Assistance (ESA), the traction control (TC), and the power transmission (PT) as shown in Fig. 13. Models can be developed for the ESA, TC and the PT sub-systems as well. To achieve a more realistic model, this work can be extended to incorporate the other closely related sub-systems and analyze the composed model for reliability/availability and sensitivity. To extend the framework established for modeling the ABS sub-system for dealing with the composition of multiple sub-systems, a new strategy would need to be developed because the model already exploits the maximum degree of complexity that is feasible to solve.

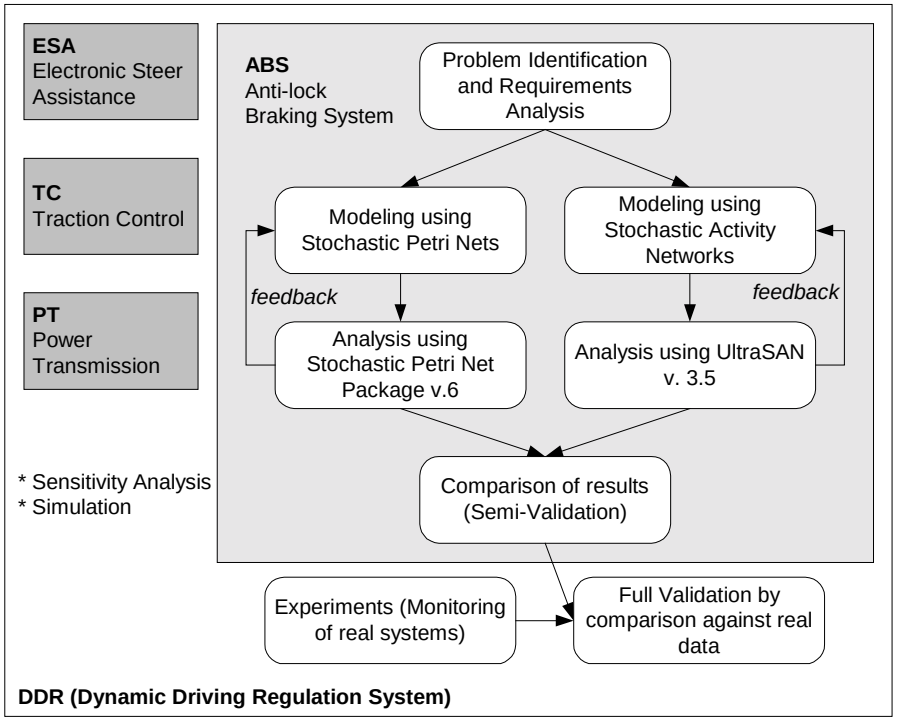

Fig. 13: Schematic showing research process and scope for future work

Both the SPN and SAN models lend themselves to sensitivity analysis at various levels. The models developed so far can be used to carry out sensitivity analysis of the system under study/development to identify the components that are most likely to fail, thereby making the

system susceptible to critical failures. Armed with such knowledge system and software architects can make more informed decisions as to the inherently reliable and safe choices and/or make economic/cost tradeoffs.

The DDR system is a very complex system, and a model capturing all its essential features/characteristics would itself be very complex, precluding any possibility of an analytical solution. In this case, the model must be studied by means of simulation i.e., numerically exercising the model for the inputs in question to determine how they affect the output measures of performance [24].

The automotive industry has several experimental studies in progress that record the effect of various system components and their failure rates on the vehicle's safety and reliability properties. The technical strategy for validation includes the measures (techniques) and procedures that would be used for confirming that each safety function conforms to the specified system safety requirements [4]. One would like to have data collected that was comprehensive enough to account for the contribution of coincident failures and usage information, data about (1) the effect of degraded operation/loss of stability on component failure rate, (2) the correlation of failures between components, (3) the effect of demand/usage on failure rates and, (4) quantization of workload durations. Such data can provide sufficient evidence to corroborate and/or validate the results obtained from these analyses or the basis for evolution of the developed models to more realistic models characterized by measurements from real experiments.

ACKNOWLEDGEMENT

We thank Juergen Schwarz (Director of System Safety), Stefan Greiner (Design to Quality Technical Lead), and Matthias Benzinger (Design Center) at DaimlerChysler for their help and

insightful comments on our work. We also thank Zhihe Zhou, a member of the Software Engineering for Dependable Systems Laboratory, for his help in proofreading this document.

REFERENCES

1. M. A. Vouk, "Software reliability engineering," *presented at the 2000 Annual Reliability and Maintainability Symposium (RAMS)*, Los Angeles, CA, 2000.
2. K. Jerath and F. T. Sheldon, "Reliability analysis of an anti-lock braking system using Stochastic Petri Nets," in *Proc. of 5th Int. Workshop on Performability Modeling of Computer and Communication Systems (PMCCS 5)*, Erlangen, Germany, 2001, pp. 56-60.
3. F. T. Sheldon, S. Greiner and M. Benzinger, "Specification, safety and reliability analysis using Stochastic Petri Net models," in *Proc. of 10th Int. Workshop on Software Specification and Design*, San Diego, CA, 2000, pp. 123-132.
4. *Functional safety of electrical/electronic/programmable electronic safety-related systems*, CEI/IEC Standard 61508, 1998.
5. F. A. Gay, "Performance evaluation for gracefully degrading systems," in *Proc. of 9th Annual Int. Symposium on Fault-Tolerant Computing (FTCS-9)*, Madison, Wisconsin, 1979, pp. 51-58.
6. M. Hecht, D. Tang and H. Hecht, "Quantitative reliability and availability assessment for critical systems including software," in *Proc. of the 12th Annual Conference on Computer Assurance*, Gaitherburg, Maryland, 1997.
7. D. P. Siewiorek and R. S. Swarz, *Reliable computer systems: Design and evaluation*. 2 ed. Woburn, MA: Digital Press, 1992, pp. 908.
8. G. Balbo, Professor, Universita di Torino, Italy. Personal Communication at EEF-Summerschool Formal Methods and Performance Analysis, Netherlands, July 2000.
9. J. B. Dugan, "Experimental analysis of models for correlation in multiversion software," in *Proc. of 5th Int. Symposium on Software Reliability Engineering*, Los Alamitos, CA, 1994, pp. 36-44.
10. D. E. Eckhardt and L. D. Lee, "Theoretical basis for the analysis of multiversion software subject to

coincident errors," *IEEE Transactions on Software Engineering*, vol. 11(12), pp. 1511-1517, 1985.

11. B. Littlewood and D. R. Miller, "Conceptual modeling of coincident failures in multiversion software," *IEEE Transactions on Software Engineering*, vol. 15(12), pp. 1596-1614, 1989.
12. J. Arlat, K. Kanoun and J.-C. Laprie, "Dependability modeling and evaluation of software fault-tolerant systems," *IEEE Transactions on Computers*, vol. 39(4), pp. 504-513, 1990.
13. J. Meyer, Professor, University of Michigan, Ann Arbor, MI. Personal Communication at PMCCS5, Erlangen, Germany, September 2001.
14. M. C. Hsueh, R. K. Iyer and K. Trivedi, "Performability modeling based on real data: A case study," *IEEE Transactions on Computers*, vol. 37(4), pp. 478-484, 1988.
15. X. Castillo and D. P. Siewiorek, "Workload, performance and reliability of digital computing systems," in *Proc. of IEEE 11th Int. Symposium on Fault Tolerant Computing*, Portland, ME, 1981, pp. 84-89.
16. J. Meyer and L. Wei, "Analysis of workload influence on dependability," in *Proc. of IEEE 18th Int. Symposium on Fault Tolerant Computing*, Tokyo, Japan, 1988, pp. 84-89.
17. L. M. Malhis, W. H. Sanders and R. D. Schlichting, "Numerical evaluation of a group-oriented multicast protocol using Stochastic Activity Networks," in *Proc. of 6th Int. Workshop on Petri Nets and Performance Models*, Durham, NC, 1995, pp. 63-72.
18. M. A. Qureshi and W. H. Sanders, "The effect of workload on the performance and availability of voting algorithms," in *Proc. of 3rd Int. Workshop on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'95)*, Durham, NC, 1995, pp. 217-224.
19. K. Jerath, "Modeling and Stochastic Analysis of Embedded Systems Emphasizing Coincident Failures, Failure Severity and Usage-Profiles," Masters thesis, Department of Electrical Engineering and Computer Science, Washington State University, Pullman, WA, 2002. www.eecs.wsu.edu/~kjerath/Papers/Jerath-Thesis.pdf.
20. G. Ciardo, J. K. Muppala and K. Trivedi, "SPNP: Stochastic Petri Net Package," in *Proc. of Int. Workshop on Petri Nets and Performance Models*, Kyoto, Japan, 1989, pp. 142-151.

21. J. Couvillion, et al., "Performability modeling with UltraSAN," *IEEE Software*, vol. 8(5), pp. 69-80, 1991.

22. F. T. Sheldon and S. Greiner, "Composing, analyzing and validating software models to assess the performability of competing design candidates," *Annals of Software Engineering*, vol. 8, pp. 239-287, 1999.

23. R. Bosch, *Automotive Handbook*. 4 ed. Manchester, Tennessee: Bentley Pubs, 1993, pp. 852.

24. A. M. Law and W. D. Kelton, *Simulation modeling and analysis*. 3 ed. New York: McGraw Hill, 2000, pp. 760.