Designing safety-critical systems: A Convergence of Technologies

Juan R. Pimentel
Kettering University
1700 W. Third Ave.
Flint, Michigan USA
jpimente@kettering.edu

Abstract – *A brief overview of the fields that must be considered when designing safety-critical systems is presented. Proper application of these fields allows a holistic (dependability achieved at all system levels) approach for designing safety-critical systems. The fields to be considered are: application domain, embedded systems, protocol and networks, safety and reliability, real-time, and systems engineering.*

I. INTRODUCTION

Designing safety-critical systems is a complex endeavor particularly if extensive use of advanced electronics and information technology is used. The increased use of microcontrollers in modern automotive systems has brought many benefits such as the merging of chassis control systems for active safety with passive-safety systems. Unfortunately, it has also brought the potential for catastrophic failures [1]. Thus, the widespread application of microcontrollers requires extreme care in order to produce a dependable system. Dependability involves reliability, safety, availability, and security but in this paper we are only concerned with safety, reliability, and availability.

The area of system safety is well-established, and procedures exist to identify and analyze electromechanical hazards along with techniques to eliminate or limit hazards in a final product. Unfortunately, much more is known about how to engineer safe mechanical systems than safe computing systems, particularly when software is a major component of the engineered system. With the increased used of software in safety-critical components of complex systems, governments agencies and other institutions are increasingly including requirements for software hazard analysis and verification of software safety (e.g., MISRA, MIL-STD-882B, IEC 61508, DO-178B). More effective modeling and analysis tools are needed to aid automotive engineers perform safety analysis of systems involving electronic and information technology components.

Safety-critical systems have many requirements that stem from several engineering disciplines. The main disciplines having a direct bearing on designing safety-critical systems are: domain engineering, embedded systems engineering, protocol and network engineering, safety engineering, reliability engineering, real-time systems engineering, and systems engineering. Focusing on one or few of these disciplines is not enough; one has to take a holistic approach that goes beyond protocols and networks, that goes beyond using a suitable microcontroller and operating system. Currently, several design and implementation options are available to a researcher, developer, or designer. In terms of protocols, one can choose among CAN, TTCAN, Switched Ethernet, TTP/C, FlexRay, and others. Because of cost, flexibility, the intended application, theoretical advances, implementation technology, and other issues, it is not straightforward to decide what protocol or network technology is the best. For example, the Microchip corporation has inexpensive sensor and actuator CAN nodes that do not require a microcontroller. In addition, it is possible to incorporate a time-triggered service on top of the CAN protocol and make it a bit more similar to TTP/C and FlexRay which are inherently time-triggered protocols. TTCAN incorporates a time-triggered service as part of the communication protocol similar to TTP/C and FlexRay. It is also possible to incorporate an event service on top of a time-triggered communication system as is the case with FlexRay. On the other hand, to produce a superior design, it is necessary to understand relevant issues from the disciplines listed above. Thus the benefits of using some protocols is becoming gray rather than black and white contributing to the complexity of safety-critical systems.

III. MAIN TECHNICAL DISCIPLINES

Only the most relevant and appropriate disciplines need to be addressed as discussed next. The criterion used is that these disciplines are at the heart of the safety-critical electronic and information technology components of modern vehicles.

A. Domain Engineering. Safety-critical systems exist in a certain application context. Certainly the details of safety-critical aerospace systems are different from those of the space shuttle, process control, or automotive. The reason that deep knowledge in a certain domain (i.e., application) is important is that it can be used to tune or optimize certain mechanisms (e.g., communication, fault tolerance, fail status, etc.). A good example is the improvement on the TTP/C group membership strategy based on the statistical distribution of periodic messages of an automotive application [6].

If one has deep understanding of the dynamics of controlled systems then appropriate values for timers can be found. In addition, keen awareness of application behaviour enables one to devise appropriate and effective error detection and recovery mechanisms that may not be optimal for all applications but certainly effective for the application in question. For many controlled systems, values read by sensors are not random -- they follow the laws of physics.

Thus it is possible to use pre-processing functions (e.g., Kalman filtering) that will help provide an estimate of the values in case of errors and faults. The above implies that knowledge of the application domain can be used to design a more dependable system.

B. Embedded System Engineering. In this paper, we assume that safety-critical systems are embedded systems and thus issues such as microcontrollers, real-time operating systems, memory configurations, and I/O are relevant. Kopetz [8] pointed out the importance of the host to communication controller interface termed *communication network interface* (CNI) and host to process I/O system interface termed *controlled object interface* (COI). In fact the TTP/C protocol has made the CNI as an integral component of the protocol. Sometimes the performance of these interfaces or the operating system will mask the performance of the underlying communication protocols. A good example is the priority-inversion problem that arise with some CAN implementations. Under certain conditions [7] a low priority message gets stuck at the transmit buffer and force a high priority message at the same node to wait indefinitely! Fortunately, there are ways to prevent this problem and some CAN implementations avoid this problem altogether.

C. Protocol and Network Engineering. Protocols and networking are at the heart of distributed safety-critical systems. In the domain of automotive engineering several protocols and networks have been used for example: J1850, J1939, and other CAN-based protocols. Other options include LIN, TTCAN (time-triggered CAN), Switched Ethernet, TTP/C, and FlexRay. The paradigms of event-triggered and time-triggered communications help understand the behaviour of these protocols. It used to be that there was a clear distinction between protocols and networks in terms of these two paradigms. The situation is not so clear now as CAN-based networks use time-triggered mechanisms at higher levels and protocols such as FlexRay offer both: time-triggered and event-triggered communication options. Another issue is the degree of flexibility offered by the protocol in order to experiment with and provide higher layer protocols (HLP). Still another issue is the application of inter-networking (using bridges, switches, and routers) at the vehicle level.

D. Safety Engineering. Whereas *system (availability) reliability* deals with the problem of ensuring that a system performs a required task or mission (at) for a specified time, *system safety* is concerned with ensuring that a *mishap* does not occur in the process. Usually, there are some failures that only cause a benign interruption of the system services while other failures cause catastrophic interruptions. The former are called *benign failures* while the latter are called *catastrophic failures*. A mishap[1] is an unplanned and undesirable event or series of events that could result in injury, illness, death, or damage to or loss of property or equipment. A *hazard* is an undesirable condition that has the potential to cause or contribute to a mishap. The situation that results from the occurrence of a mishap is called a *failure mode*. Thus according to these definitions, a mishap occurs when the conditions of a hazard are fulfilled. Hazards and mishaps can be classified in various severity levels ranging from negligible to catastrophic.

More specifically, hazards can be categorized by the aggregate probability of the occurrence of the individual conditions that make up the hazard and by the seriousness of the effects of the resulting mishaps. Together these constitute risk. More specifically,

$$risk = h_s\, h_p \qquad (1)$$

Where h_s is the hazard severity and h_p is the hazard probability. Eq. (1) applies for a single risk rather than for all risks.

Because of the severity and probabilistic aspects of risk, there are two major views of safety: probabilistic and system [1]. The system view is more practical and holistic that includes all remaining non-probabilistic issues. The first step in a safety analysis is to identify the system hazards and assess their severity and probability (i.e., risk). The vast majority of available safety tools and methods support severity analysis [2,3].

The overall goal in designing a safety-critical system is to eliminate hazards from the design of (if that is not feasible) to minimize risk by modifying the design so that there is a very low probability of the hazard occurring. To demonstrate that a system is safe, it is necessary first to ensure that given that the specifications are correctly implemented and no failures occur, the operation of the system will not result in a mishap. Second, the risk of faults or failures leading to a mishap must be eliminated or minimized by using fault-tolerant or fail-safe procedures. If it is not possible to completely eliminate hazards, then in order to reduce risk, the exposure time (length of time of occurrence) of the hazard must be minimized.

Hazards need to be handled appropriately whether they result from component failures or whether they exist as a result of other failures such as those due to design errors or unforeseen events. The following is a list (not exhaustive) of control failures that need to be considered in a system safety analysis:

- A required event that does not occur
- An undesirable event
- An incorrect sequence of desired events
- Two incompatible events occurring simultaneously
- Timing failures in event sequences
- Exceeding maximum time constraints between events

E. Reliability Engineering. Reliability engineering deals with the continued or available operation of a system even under the failure of system components. The primary mechanism is the use of redundant components to design fault-tolerant systems; a system that continues to provide

[1] To include harmful exposures, e.g., toxic, the Deapartment of Defense uses mishap in terms of accident [1].

services perhaps with a degraded level of performance when some of the component fail. There are two well known schemes to handle the replacement of failed components: static redundancy and dynamic redundancy. Just as in the case with safety, there are two major views of reliability: probabilistic, and system. The probabilistic analysis yields availability and reliability functions in terms of component failure rates. From these functions, one can easily evaluate the MTTF (mean time to system failure) and MTTR (mean time to repair).

F. Real-Time Engineering. Techniques for ensuring that a system meet timeliness requirements are important for safety-critical applications. A distinction is made between hard real-time and soft real-time systems. The former are systems that will result in grave consequences if they do not meet real-time requirements and the latter imply that the consequences are not so grave. Safety-critical systems certainly belong to the category of hard real-time systems. To see if a system meets real-time requirements, schedulability analysis is used and this methodology is well known for single-processor or multiprocessor operating systems. Schedulability techniques have been extended to CAN communication protocols [9] and others.

G. Systems Engineering. Systems engineering emphasizes formal processes that start with a system's requirements and specification, and includes an iterative design, test, and verification cycle. A good example of a system process used by the automotive industry is the V cycle. For safety-critical systems the emphasis is on using a safety process rather than specifying established techniques for ensuring safety, reliability, etc. The process emphasizes a plan and various assessment levels to relate a finished implementation to the requirements and specifications.

Several automotive associations and institutions (e.g., MISRA) are concerned with guidelines, techniques, and processes to deal with hazards and safety issues. A key aspect of these guidelines is that the hazards associated with a system must be both understood, and taken into consideration, from the beginning of its design cycle. The following has been identified as important [5]:

- Assess the risks associated with the behaviour of a system;
- Do this early enough in order to take design actions that can reduce those risks to an acceptable level;
- Provide documentary evidence of the reasoning that lies behind the design decisions made.

A number of processes for safety analysis exist or have been suggested [4] that involve the following stages: preliminary safety analysis (PSA), preliminary hazard analysis (PHA), detailed safety analysis (DSA), and safety integrity levels (SIL). An example of a process used in the aerospace industry is DO-178B. The automotive industry is expected to use or develop a similar safety process.

III. DISCUSSION

Designing safety-critical systems is a complex endeavor involving several fields as discussed above. This explains why much more is known about how to engineer safe mechanical systems than safe computing systems. For safety-critical systems the emphasis is on using a safety process rather than specifying established techniques for ensuring safety, reliability, etc. New concepts in automotive engineering (e.g., x-by-wire) as well as distributed control systems require time-triggered-communication. As we progress in the protocol/network development, a holistic approach to designing safety-critical systems becomes necessary. Protocols are evolving to include more common features (e.g., time triggered on top of event triggered and conversely). Researchers, developers, and designers have several options for designing safety critical systems. Thus the benefits of using some protocols is becoming gray rather than black and white contributing to the complexity of safety-critical systems.

Several of the issues raised in this paper are a result of observations from several on-going experimental projects at the distributed embedded systems laboratory at Kettering University.

IV. REFERENCES

[1] Leveson, N.G., *Safeware*, Addison-Wesley, 1995.

[2] Hammer, W., *Handbook of System and Product Safety*, Prentice-Hall, 1972.

[3] Vesely, W.E., Goldberg, F.F., Roberts, N.H., and Haasl, D.F., Fault Tree Handbook, NUREG-0492, U.S. Nuclear Regulatory Commision, Jan. 1981.

[4] Amberkar, S., D'Ambrosio, J.G., Murray, B.T., Wysocki, J., and Czerny, B.J., "A System-Safety Process for By-Wire Automotive Systems," SAE Congress paper, 2002.

[5] Jesty, P.H., Hobley, K.M., Evans, R., and Kendall, I., "Safety Analysis of Vehicle-Based Systems," Proceedings of the 8th Safety-critical Systems Symposium, 2000.

[6] Latronico, E., and Koopman, P., "A Period-Based Group Membership Strategy for Nodes of TDMA Networks," FeT03, July 2003.

[7] Pimentel, J.R., "Modeling the Response of an Automotive Event-Based Architecture: A Case Study," SAE 2003 World Congress, Paper No. 2003-01-1199, Detroit, MI., March 2003.

[8] Kopetz, H., *Real-Time Systems, Design Principles for Distributed Embedded Applications*, Kluwer Academic Publishers, 1997.

[9] Tindell, K., "Analysis of Hard Real-Time Communications," Real-Time Systems, Vol. 9, pp. 147-171, 1995.