2005

Representing Design Tradeoffs in Safety-Critical Systems

Jennifer Black
Carnegie Mellon University, jenmorris@cmu.edu

Philip Koopman
Carnegie Mellon University, koopman@cmu.edu

Published In
.

Representing Design Tradeoffs in Safety-Critical Systems

Jennifer Morris
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213 USA
jenmorris@cmu.edu

Philip Koopman
Carnegie Mellon University
5000 Forbes Ave.
Pittsburgh, PA 15213 USA
koopman@cmu.edu

ABSTRACT
Different fault-tolerance strategies have been shown to be effective at achieving fail-safe behavior in a number of safety-critical application domains with different dependability, service, and cost requirements. A technique for comparing the domain profiles and their fault-tolerance strategies could assist architects of new safety-critical systems in choosing an appropriate fault-tolerance strategy. We suggest an approach using Kiviat graphs to visually represent the dependability, service, and cost profile of a system, and show how such a graph can be used to analyze automotive x-by-wire applications.

Categories and Subject Descriptors
D.2 [**Software**]: Software Engineering; D.2.11 [**Software Engineering**]: Software Architectures—*Domain-Specific Architectures*

General Terms
Design, Reliability

Keywords
Safety, dependability, availability, integrity, fault-tolerance, modeling

1. INTRODUCTION
Safety-critical systems increasingly rely upon computerized controls to provide advanced services and safety features that are impracticable to realize in hardware alone. Meeting the safety and service requirements of these software-intensive systems is challenging, not only because they are often more complex than their electromechanical predecessors, but also because their dependability and safety requirements are often greater [4].

Safe software-based system design has been successful in some application domains where safety engineering for electromechanical systems has previously been well established. Both the rail and commercial aviation industries use standard fault-tolerance techniques to minimize critical system failure rates to meet industry-determined standards. The techniques they choose, however, may vary according to the safety, service, and cost requirements of the particular system being designed. In railroads, signaling and switching systems that regulate traffic often use dual two-of-two redundancy because they have very high service availability demands over a long mission time. In commercial aircraft flight control systems, reliability and availability are also very high, but with a much shorter mission time. Therefore, the fault-tolerance level necessary to meet the safety and service requirements of a flight control system can often be achieved through triple-modular redundancy.

As other safety-critical application domains transition to greater reliance on software, a natural reaction of system designers might be to simply apply the same safety-critical strategies used in these well-established systems. If fault-tolerance strategy X works for application A, why not use it in application B? The problem with this approach is that it often only takes into account the reliability and availability requirements of the new and existing systems, without considering differences in other system requirements. For example, strategy X may be too expensive to implement or excessive in application B. Also, safety requirements depend on the application domain.

A visual representation of high-level system requirements might be a useful tool for system designers trying to see the "big picture" of the application domain. There are a number of ways to model the detailed behavior of specific implementations, but relatively few tools for comparing application domain properties prior to system design. In this paper we propose a new technique that uses Kiviat graphs to profile system dependability, service, and cost requirements. We then demonstrate how these graphs can be used to consider safety design strategies for automotive X-by-wire applications. This work is not a proof of concept or implementation, but rather is a suggestion for a way to think about domain-specific requirements in safe system design.

2. SAFETY-CRITICAL SYSTEM PROFILES
Kiviat graphs, originally proposed by Kolence and Kiviat [3], have traditionally been used as a technique for comparing software performance profiles [2]. Different system characteristics or parameters, such as CPU and channel utilization, are plotted on radial axes to form a profile of system

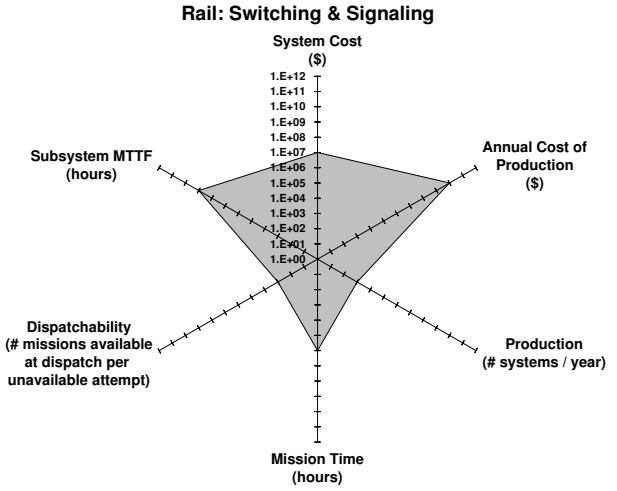

Figure 1: Kiviat graph for rail signaling & switching system

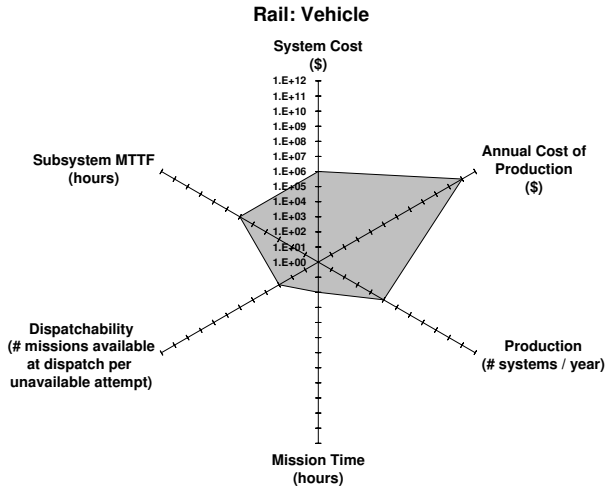

Figure 2: Kiviat graph for a rail car

behavior. System architects can compare the profiles of different systems to reason about their observed and expected behavior. Kiviat graphs can also provide some insight into application domain-specific requirements for safety-critical systems.

Figures 1 and 2 show Kiviat graphs of the domain requirements of a rail signaling and switching system and a rail car. Although there may be additional system characteristics that could provide additional insight for system architects, in this analysis we choose to focus on six different system parameters that impact system design choices: individual system cost, total annual production cost, annual production, mission time, dispatchability, and reliability expressed as a mean time to failure (MTTF). These numbers are approximations based on industry estimates and used solely to make order-of-magnitude comparisons.

The individual system cost, annual production, and total annual cost (which is the product of the first two parameters) refer to the encompassing system (e.g., the aircraft, the rail vehicle, the rail control infrastructure, etc.). The total annual cost of production can be considered a general approximation of the upper-bound for software costs, since these are amortized over all units produced. Likewise, the individual system cost can approximate the upper bound on the hardware cost of the system. In other words, if the system were entirely composed of hardware then the system cost would be the hardware cost. If the system were entirely composed of software, the entire cost of producing all systems would equate to the software cost.

The mission time is the expected length of operation of one mission for the system. Dispatchability and MTTF are dependability measures, where dispatchability refers to the availability to begin a mission (i.e., no faults have occurred in the system at the time the mission starts), and MTTF is the reliability requirement of the system.

The following two sections describe the domain profiles of two domains often cited as examples of software system safety: rail and commercial aviation. The different safety, service, and cost requirements of these two domains have resulted in different approaches to fault-tolerance. The third section compares the profiles of these systems to two possible domain profiles for automotive x-by-wire applications to identify similarities that can help guide architectural design decisions in these new systems.

2.1 Rail

The rail industry has been involved in safety-critical system design long before the introduction of computers. In rail systems, the greatest hazards involve collisions between trains. Although careful scheduling of routes can avoid many opportunities for hazards to occur, a safety system is needed to identify and prevent unexpected collisions from occurring. Traditionally, this safety system consisted of human operators to manually switch tracks and signal safe routes and mechanical interlocks to physically prevent the operators from doing otherwise. Examples of mechanical interlocks include physical connections between the track and the signals that prohibit unconnected rails from being marked as clear, and locking bolts to prevent track switching under a stationary car.

Over the years, railways have turned to computer control systems to reduce the reliance on human operators and provide more efficient operation. Advancements in hardware (processors, sensors, actuators, etc.) and software allow more complex scheduling of routes to better utilize track resources. Although some mechanical interlocks remain in these systems, many systems also rely on software-based systems to prevent hazards.

In the event of a system fault, a safe failure state for the signaling and switching system could be the cessation of all railroad traffic. If no cars are allowed to move, then there would be no possibility of a collision between cars. In this case, a fault-tolerant design could be a single two-of-two system. However, service requirements of the system are much

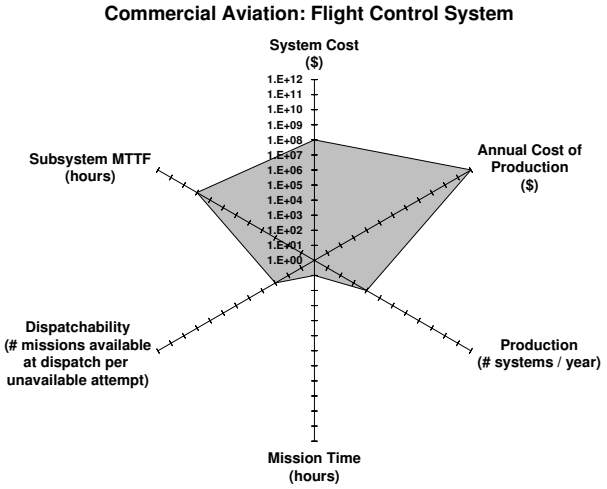

Figure 3: Kiviat graph for commercial flight control system

higher. Often, such systems are expected to operate continuously for tens of years, despite failures of individual components. For this reason, such systems are typically designed as dual two-of-two systems. When one two-of-two system detects a fault, operations can continue with the other while the faulty component is repaired.

The rail cars themselves, which are also generally safe when physically stopped, do not have to meet the availability standards of the signaling system and are typically built using a single two-of-two system. This is important because in general each signaling and switching system corresponds to several individual cars. The signaling and switching system of the Bay Area Rapid Transit System (BART)in California controls approximately 600 cars [1]. Recently the Advanced Automatic Train Control system was upgraded at a contracted cost of $45 million. The additional cost of the dual two-of-two system is more easily managed when the cost of the overall system is high and the production is low. The cost of a car for the BART system is approximately $2 million. Note in the Kiviat charts for each system, although the individual unit cost of the signaling and switching system is higher than that for the cars, the overall total production cost of the cars is an order of magnitude higher due to the higher production numbers. This is interesting because it may indicate that although the signaling and switching system can afford higher hardware costs, it may be more restricted on software costs than rail cars.

2.2 Commercial Aviation

Commercial aviation has also cultivated safe system design before and after the introduction of software-based systems. Like rail signaling and switching systems, the reliability requirements for avionic flight control systems are also quite high, with MTTF on the order of 10^{-9} failures per hour. In contrast to rail control systems, flight control systems must continue to operate in the presence of system faults. Stopping an aircraft abruptly in mid-flight is not a viable fail-safe strategy. Therefore, high reliability requirements are mandated by system safety considerations, not just service requirements. The mission time over which aircraft must operate without critical failures is a relatively small ten or twenty hours. A Kiviat graph of the domain requirements for aviation flight control systems is shown in figure 3. A comparison of this domain profile with that of the rail switching and signaling system indicates that the requirements of both domains are similar, with exception of the mission time which is much higher for the rail control system.

To achieve reliability requirements, aviation controls are often triple-modular redundant. In the event of a fault, the system continues to operate with full service until the end of a mission. Repair of the faulty component is not required during the mission, but must, in most cases, be completed before the next mission for dispatchability. For commercial aircraft an average flight cancellation rate due to all sources is perhaps 1-2% [5]. It seems reasonable to assume that the flight cancellation due to failed redundant units in the aircraft should be no more than one tenth of this (0.1-0.2%).

Triple-modular redundant systems tend to have high unit costs because they require extra hardware. For aircraft, this cost is acceptable because the overall system cost is also high (hundreds of millions of dollars). It is interesting to note that both the total annual cost of production and the individual system cost of commercial aircraft are each approximately ten times greater than the same values of a rail signaling and switching system. This may indicate that the cost of both hardware and software in commercial aircraft can reasonably be expected to be greater than that of rail control systems.

2.3 Automotive X-by-Wire

The automotive industry has recently begun transitioning safety-critical subsystems to computerized control. Historically these systems have achieved safety through mechanical back-ups. However, the potential for new features and reduced cost have motivated the elimination of these backups in favor of systems that are completely computer controlled (X-by-wire). The problem with removal of physical safety mechanisms in these systems is that the underlying com-

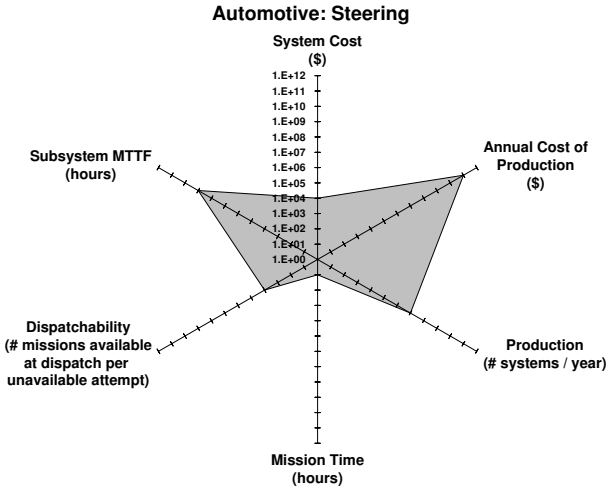

Figure 4: Kiviat graph for automotive steer-by-wire (and brake-by-wire w/o backup)

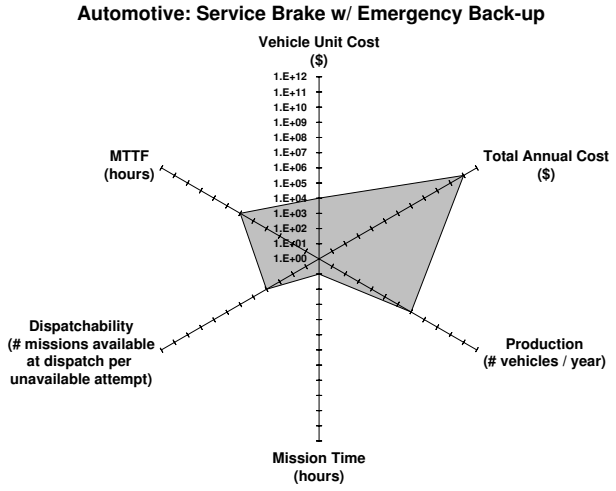

Figure 5: Kiviat graph for automotive throttle-by-wire (and brake-by-wire w/backup)

puter architecture often does not contain adequate fault-tolerance to maintain system safety alone. For that reason, it may be necessary to architect some automotive X-by-wire subsystems to include more robust fault-tolerance.

The domain profiles of two types of automotive subsystems are presented in figures 4 and 5. The first, a steer-by-wire system, must provide continuous service in the presence of system faults to maintain system safety, and therefore has very high reliability requirements, on the order of 10^{-9} - 10^{-10} failures per hour. It is also reasonable to assume that consumer demands place a much higher dispatchability requirement on cars than on planes, given that the number of replacement vehicles available to an individual driver is much lower. We estimate an expected mission cancellation rate due to system faults should be approximately ten times lower than for aircraft (0.01-0.02%).

The second, a throttle-by-wire system, can remain safe after a component fault by stopping operation. A car in motion when the throttle is lost will not grind to an abrupt stop, but rather will gradually slow down, often allowing time for the driver to maneuver to the side of the road. Although it is possible to come up with extreme scenarios where this might be problematic, it is reasonable to consider this a "safe" system behavior. Therefore, the throttle might only require a MTTF of 10^{-6} failures per hour.

In the case of brake-by-wire, a separate parking/emergency brake may exist to provide a mechanical back-up for the service brakes. The idea is that if the brakes fail while the vehicle is moving, the driver can still engage the parking/emergency brake to bring the vehicle to a stop. In this architecture, both service brake and parking/emergency brake profiles might resemble that of the throttle. If a separate parking/emergency brake is not available, the brake-by-wire subsystem profile would be similar to that of the more critical steer-by-wire subsystem.

When the domain profiles for the two types of automotive X-by-wire systems are compared with those of the rail and aviation applications, it is clear that automotive domain requirements do not map one-to-one with either system. The steer-by-wire system has reliability and mission time requirements that are quite similar to the aviation flight control system. This may indicate that similar fault-tolerance mechanisms (i.e., triple modular redundancy) is necessary to maintain safety in these systems. However, the Kiviat graphs also indicate that the unit cost of automobiles is much smaller, and the production numbers much higher than those of commercial aircraft. This increases the overall impact on system cost of the hardware redundancy needed for TMR. For the automotive x-by-wire applications with lower reliability requirements, some of the less expensive techniques used in rail cars may be more appropriate.

Another interesting observation stemming from the Kiviat graphs is that the total annual cost of the flight control system is ten times greater than the total cost of the rail and automotive domains. If we assume that the total software budget cannot exceed this amount, it is possible that the maximum allowable expenditure on safety-critical software for automotive and rail applications might be much lower than that used in commercial avionics. A triple modular redundant flight control system with voting is likely to be more complex, and therefore more costly to verify and validate, than two-of-two fail-silent systems in rail cars or dual two-of-two fail-operational in rail signaling and switching. For automotive applications with service and dependability demands similar to aviation applications, this may require limits on system complexity to reduce software development, verification and validation costs.

3. CONCLUSIONS

This paper has presented a possible technique for comparing application domain properties and demonstrated its use in the comparison of three general safety-critical application domains. The main purpose of such a tool would be to provide system architects with a structured way of comparing attributes of different application domains prior to

system design. Future work in this field would require a more thorough definition of the parameters to be analyzed, more detailed analysis of specific systems, and verification of the results of analysis.

4. ACKNOWLEDGMENTS

This material is based upon work supported under a National Science Foundation Graduate Research Fellowship. Any opinions, findings, conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

5. REFERENCES

[1] BART. *BART System Facts*. San Francisco Bay Area Rapid Transit District Website, http://www.bart.gov/about/history/systemFacts.asp, accessed February 28, 2005.

[2] M. Esponda and R. Rojas. A graphical comparison of risc processors. *ACM SIGARCH Computer Architecture News*, 20(4):2–8, September 1992.

[3] K. W. Kolence and P. J. Kiviat. Software unit profiles & Kiviat figures. *ACM SIGMETRICS Performance Evaluation Review*, 2(3):2–12, September 1973.

[4] N. Leveson. *Safeware: System Safety and Computers*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1995.

[5] U.S. Department of Transportation. *Air Travel Consumer Reports*. U.S. Department of Transportation Website, http://airconsumer.ost.dot.gov/reports/index.htm, accessed February 28, 2005.