COMPLEXITY *by* DESIGN

As engineered systems grow, it's imperative to understand them well enough to forestall unexpected failures.

We encounter engineered systems every day, whether we realize it or not. The logistics that deliver your cup of coffee or the technology that enables you to place a call on a smart phone are incredibly complex.

And yet, most people don't take time to consider the workings of these systems. Either they work—and are thus ignored—or they fail and are seen as too opaque to understand.

Engineers don't have the luxury of ignorance. We who design these complex systems have to understand how the various components of a system fit together and anticipate how the interactions between these components could lead to failure. Instead of shrugging at the opacity of the technology, engineers have to use tools that make the inner workings transparent. Those tools may be bits of technology—or they may be bodies of knowledge that engineers tap into.

By Shannon Flumerfelt, Gary Halada, and Franz-Josef Kahlen

Complex systems come in all sizes. For instance, consider the system that routes customer baggage through a modern airport. In such a system, users rely unquestioningly on an almost inconceivable level of sophistication.

Twenty years ago, after a bag entered the system at the check-in counter, its exact whereabouts in the baggage system running on conveyor belts was largely unknown. Locating a particular piece of luggage in transit between planes or from a plane to the baggage claim could take several hours. This level of uncertainty reduced the ability of airlines to schedule connecting flights with less than a 45-minute window to transfer bags, and even then, lost luggage was a fact of air travel.

To speed luggage transfer and reduce uncertainty in the location of bags, new systems were designed in the 1990s that combined high speed carts to replace conveyor belts and automated barcode readers. One of the first large-scale experiences with such system came in 1993 when the Denver In-

Shannon Flumerfelt is an endowed professor of Lean and director of Lean Thinking for Schools at The Pawley Lean Institute at Oakland University in Rochester, Mich. Gary P. Halada is an associate professor in materials science and engineering at Stony Brook University in New York. Franz-Josef Kahlen is an associate professor in the Department of Mechanical Engineering at the University of Cape Town, South Africa.

ternational Airport opened up. The scale of the airport required a new type of baggage system, but this did not become apparent until just two years before the facility was to open. The baggage system had to accommodate the existing arrival and departure hall architecture rather than have those be built around the system, and the short time table meant that the system was not validated and that no backup system was prepared.

The first test runs of the newly installed, automatic, remote barcode readers were a disaster; luggage was delivered to other flights, disappeared totally, or was shredded. Indeed, the opening of the airport was pushed back by more than one year but the automated baggage system never worked as designed and was finally decommissioned 10 years after the airport opened.

Another, more successful example of a complex system is the computer hard drive onto which digital data are stored. These are a marvel of electromechanical integration: Data are written onto disk drives in narrow tracks by a magnetic recording read/write head held in place by a suspension system capable of maintaining a stand-off distance to the actual disk of less than 10 nm.

The complex design parameters for these suspensions in disk drive systems involve at least two levels that computer users generally don't appreciate. First, because disk drives operate at several thousand RPMs, the data must be written to and read from the disk drive quickly, all the time, every time. Considering the disk drive diameter and typical RPMs, and the requirement to maintain a stand-off distance to the actual disk of less than 10 nm at all times, the task of the suspension can be scaled up to a Boeing 747 flying just two inches off the ground.

Because disk drives operate at several thousand RPMs, the data must be written to and read from the disk drive quickly, all the time, every time. Considering the disk drive diameter and typical RPMs, and the requirement to maintain a stand-off distance to the actual disk of less than 10 nm at all times, the task of the suspension can be scaled up to a Boeing 747 flying just two inches off the ground.

The second level of complexity is introduced by aerodynamic considerations inside the disk drive. The angular velocity at the edge of the SCSI disk drive can reach Mach numbers of around 0.4 to 0.5. The design parameters for the suspension then must account for torsional stiffness against aerodynamic effects as well as longitudinal bending. Obviously, minimal miscalculations of the torsional stiffness or the longitudinal bending will lead to unreliable data storage on the disk drive.

Given those conditions, it's a triumph that hard drive manufacturers can mass produce these miniature complex systems with a failure rate of just four per million.

How do engineers design complex systems to remain reliable? Some of the most important tools for system design and optimization have been computer-aided engineering or computer-aided design. Engineers have used CAE or CAD tools extensively for the past forty years to draw and more recently simulate and test engineering designs. The development of commercial graphics software starting in the 1960s revolutionized engineering design in large companies in the automotive, defense, and aerospace industries, but the impact was limited. Early computer mainframes and their accompanying software and input devices were expensive and required detailed training. Thus, at first, only the largest companies developing complicated engineered systems could justify using these computer-based design systems.

Throughout the 1970s and 1980s, professionals in both academia and the private sector worked to create more-advanced packages capable of handling the increasingly complex demands of industry. Simple stick figure drawings were eventually replaced by surface modeling and textures, and then by solid modeling. By the late 1970s, solid models would serve as the basis for computer-assisted milling machines to produce real objects. In time, the personal computer revolution "opened the doors of the drafting room" and integrated CAE into every aspect of the design process.

Computers also revolutionized the process of mechanical and electrical analysis of engineered structures and circuits, starting with analog computers in the 1930s and 1940s at MIT. Those early systems could analyze the forces in structures with 200 to 2,000

degrees of freedom, sufficient for simple systems and components but not for real world, complex problems.

Finite element methods, developed in the 1950s and 1960s, could do much more—breaking a large computer model of an object into many small elements and then using mathematical expressions for physical properties, solved at the "nodes" between elements, to solve for overall "systemic" responses to a stress or a load at a particular location. That technique found immediate application in the aerospace industry where it was critical to understand structural responses to stress. By the late 1970s, the integration of FEM and graphical drawing and modeling software resulted in an improved way to design and test simultaneously in order to enhance performance and hence reliability.

Software can be used to train engineers in operation of systems—as long as the computing speed and power are sufficient to create a realistic simulation. If the information used to create the simulation is insufficient—or wrong—the predicted operation, including the result of any faults or overloads or other extreme conditions, will be inaccurate.

But to expand this concept to complex systems, software had to be developed to simulate a wide range of responses, not only in the case of a static stress at a point, but also for motion, transmission of signals and energy, and other dynamic input. Engineers needed to visualize how a system would respond to variations in design criteria, how it would operate in whole or in part. Programs like ABAQUS and ANSYS (both developed in the 1970s) were designed to allow engineers to expand on FEM and CAD to be able to solve so-called multiphysics problems, involving mechanical, thermal, electromagnetic and vibrational forces in a design. Software for electronics simulation and fluid dynamics and electrical control, such as Simulation Program with Integrated Circuit Emphasis, or SPICE, followed. Using computational fluid dynamics programs, designers could not only understand fluid flow through a complex system, but also could simulate surges in pressure, rapid temperature changes, and other dynamic changes that can affect overall system performance.

In recent decades, as the computing power of desktop systems expanded, creating simulations of motion, forces, flows, and operation became a commonplace tool for design engineers. They now had the ability to simulate the operation of electronic circuits through readily available programs, analyze forces resulting from impact or stresses using finite element modeling, and observe the flow of water, power, or information through pathways. Control systems could be modeled with popular mathematical software combined with graphical block diagramming software that can be used to link components and functions together to create dynamic simulations.

Such software packages can also be used to train engineers in operation of systems—as long as the computing speed and power are sufficient to create a realistic simulation of the engineered system. That's a critical requirement: if the information used to create the simulation is insufficient—or wrong—the predicted operation, including the result of any faults or overloads or other extreme conditions, will be inaccurate.

Control system design and modeling tools can do more than simply provide flow and analysis functions. Software designers and engineers have begun to incorporate functions into those applications specifically to limit the possibility of failure. For example, a new software application for designing energy production systems also allows engineers to test what-if scenarios, such as what might happen during a transient power spike that might cause system stress and failure or how the entire system responds to increasing demand over time.

The development of sophisticated expert system software that can provide rapid and intuitive access to vast amounts of data on materials and design features of available components also enables an individual engineer to tap into the expertise of many others. This is critical in helping to avoid failure from lack of knowledge.

Some automated control and feedback systems use embedded sensors and extremely rapid response mechanisms to prevent or limit damage from a failure far faster than a human operator could. This is, in a

sense, a biomimetic model for damage control, as most biological systems—including humans—constantly undergo stresses, damage, and corrective actions on a molecular level. Failure of these automatic detection and repair processes lead to many, if not all, diseases. Engineers have begun incorporating such "self-healing" processes, involving embedded detection, feedback, and correction, in a number of systems.

Many computer assisted design, operation, and maintenance tools have been developed over the past two decades. These can provide both designer and on-site engineer with key knowledge to help limit the possibility of failure. But these tools are only as good as the data they have access to, and engineers and engineering companies must make a conscientious and thorough effort to ensure that, to the best of industry standards and available knowledge, the information provided by a program is correct and appropriate to the task at hand. Results should be crosschecked if possible, with additional calculations, and software should be chosen with the need for reliability and accuracy held paramount.

Computer simulations and automated control systems are not the only means for engineers to reduce the potential for failure. When considering how to mitigate the likelihood of failure in engineering, and in complex systems in particular, the proper use of comprehensive probabilistic risk assessment is critical.

As defined by Michael Stamatelatos of the National Aeronautics and Space Administration, probabilistic risk assessment is "a systematic and comprehensive methodology to evaluate risks associated with every life-cycle aspect of a complex engineered technological entity from concept definition, through design, construction, and operation, and up to removal from service." PRA has proven to be extremely valuable in a host of complex engineered systems, ranging from chemical processing facilities and nuclear power plants to waste storage and treatment facilities and aerospace missions and devices.

The process of performing PRA, in particular during the design phase of a system, mirrors the engineering design process itself. It introduces key factors which need to be taken into account during concept generation and selection. While a number of representative equations have been developed to express what is most important in developing a PRA of an engineered system, there are three primary factors:

- The probability of failure, usually the likelihood of failure of individual components combined to express the degree of vulnerability or risk of failure for an entire system; multiplied by
- The degree of loss or magnitude of severity of the consequences of failure, and divided by
- The degree of preparedness or nature of preventive measures put into place.

As a formula, this is often written,

$$R \propto \{\sum_{1}^{n} (Pf_n \times M_n)\} \div Prep\,(sys)$$

where R is the overall risk from failure, Pf_n is the probability of failure of an individual component or subsystem (in a system composed of n independent components or subsystems), M_n is the magnitude or severity of consequences of failure of that component (in relation to the entire system), and $Prep(sys)$ is a general term related to what measures have been taken to enhance preparedness for system failure.

Obviously, a three-factor equation for risk involves vast simplification, and it incorporates an assumption that all components of the system are independent from one another. In reality, failure of one component most likely leads to failure of others. Both intentional and unintentional interactions often exist among components, subsystems, and functions, and these interactions often lie at the heart of failure mechanisms in complex systems.

Further, the concept of "preparedness" is a somewhat abstract one and requires an engineer to make many assumptions about future use, possible site

Failure of one component most likely leads to failure of others. Both intentional and unintentional interactions often exist among components, subsystems, and functions, and these interactions often lie at the heart of failure mechanisms in complex systems.

risks, and even human nature. Preparedness or precautionary activities may also be looked at in the context of risk management as well as assessment. Juergen Weichselgartner, a researcher at the Institute for Coastal Research in Geesthacht, Germany, and an expert on natural disasters, includes preparedness, physical hazards, degree of exposure of individuals and infrastructure, prevention, and response all as the key factors which define the degree of vulnerability. Weichselgartner further defines preparedness as "all precautionary activities and measures which enable rapid and effective response to hazard events."

In general, useful and accurate application of PRA to complex systems must take into account characterization of uncertainty. This uncertainty arises from both random, or probabilistic, causes as well as from uncertainty in the appropriateness and nature of the model used for PRA. As has been found in many cases of failure, uncertainty due to human factors, including behavioral, psychological, and organizational is both difficult to quantify and hard to predict.

Adaptive risk management structures, such as those used in high-reliability organizations, which rely on expertise, planning, and communication, can help to reduce the uncertainty of human factor risk. Embedded computer systems and sensor networks can also reduce the time required for a corrective response to a failure or an out-of-control process, also reducing risk due to human factors. More accurate models of systems based on expert system programs with broad and deep knowledge bases of expertise and ever improving "intelligent" inference engines also can serve to reduce risk from uncertainty due to use of inaccurate models for risk assessment and management.

There are other ways to improve the human side of risk management and to avoid catastrophic failure in complex systems. One is to focus on the development of knowledge within individual engineers and to enhance the capacity for this within organizations via knowledge management.

Knowledge development begins with trying to understand the system and how it relates to other systems and the outside world. Systems operate under the principle that the sum of the interdependent elements holds inherently different characteristics and outcomes from those of the individual elements; this is the crux of complex adaptivity. In other words, one may understand elements of a system and have the ability to respond to the state of those individual elements, but a systems approach requires the ability to envision and grasp all of the elements and their synergistic properties as holistic thinking.

The better that engineers understand the complex system they are designing or working on—that is, whether elements are missing or defective, not properly interfacing or misaligned, or not performing correctly or adapting to subpar metrics—the more they can minimize the potential for failure. This is as true for organizations as it is for individuals.

Dealing with the complex adaptivity of systems leads to a deeper understanding of risk management of systems, not as a trade-off position between redundancy and efficiency, but to treating redundancy and efficiency as possible interdependent system attributes that may serve concurrently as barriers or enablers to system performance. Redundancy and efficiency can be considered as both confounding and stabilizing to a current system's state of resilience, depending on the condition of complex adaptivity at the time.

To do this, engineers have to examine the system's elements, map the state of interdependence between and among those elements, and measure the metrics that the elements and their interrelationships produce in order to realize a system view of operations. The better that engineers understand the complex system they are designing or working on—that is, whether elements are missing or defective, not properly interfacing or misaligned, or not performing correctly or adapting to subpar metrics—the more they can minimize the potential for failure. This is as true for organizations as it is for individuals.

The rise of complex systems creates a challenge to traditional ways of engineering. Fortunately, we are developing the tools—both as technology and as professional practices—that can meet this challenge. ■