A practical guide to assuring the system resilience to operational errors

Extended abstract – July 2016

Avigdor Zonnenshain and Avi Harel

This extended abstract documents a project of the Gordon Center for Systems Engineering at the Technion. The goal of this project was to develop a guide for system engineers, with guidelines for assuring safe interaction between the operators and the machine. The guide was announced on the INCOSE Annual International Symposium (Zonnenshain & Harel, 2015, download).

There are many explanations for the source of system failures. Few of them are: organizational factors, interaction between inevitable failures, extreme operational conditions, human errors, quality of requirement specification, quality of the implementation, and mismatch with the operational context. The guide does not tackle the various sources and explanations for failure. Rather, it focuses on setting defences against common failure modes, described in a model of resilient operation.

An event that instigates an incidence is called a trigger. Common types of triggers include: an external event, a hardware unit or a component failure, power failure, such as due to battery change or weak connection, communication interference or failure, unsupported state transition due to missing specifications, design mistakes, software bugs or poor re-engineering, operator's error or mistake, such as inadvertent or deliberate activation of a control or a feature, not suited to the operational procedure, and exceptional changes in production rate. Beside the common types of triggers, special triggers may be involved in the operation of specific systems (Zonnenshain & Harel, 2013, download).

A common practice for predicting incidences is by applying various trigger-based methods of root-cause analysis, such as Fault Tree Analysis (FTA), Event Tree Analysis (ETA), Failure Mode and Effect Analysis (FMEA), and Hazard Operability studies (HAZOP). Theoretically, such methods can reveal many kinds of failure modes. Practically, the effect of these methods is limited, for the following reasons:

- Accidents in operating risky systems, such as nuclear power plants, are due to their interactive complexity. Root-cause analysis conducted manually by human beings cannot handle all the combinations of concurrent faults. The incidences materialize mainly combinations that were skipped in the analysis.

- The same sequence of events may lead to both success and failure. Therefore, root-cause analysis is useful only when going backward, by backtracking prior incidences, but is useless for predicting future incidences.
- Mishaps might be the result of incomplete system specification.
- Black Swan accidents cannot possibly be predicted, because there is no data about prior events.
- Zonnenshain and Harel (2009, download) demonstrated that often, operational errors are the result of inconsistency in the state of the extended system, and especially, of the machine-operator interaction.

Therefore, other kinds of techniques should be employed to predict risky situations.

A primary source of critical system malfunction is often attributed to human errors. Human errors explain most accidents in the air, sea, driving and in the industry. Human errors are the primary source of operational loss: by accidents, damage to property, low productivity, or user dissatisfaction. Many of the usability issues in operating consumer products are due to use errors (Zonnenshain & Harel, 2009, download). Yet, the meaning of the term "human error" is ambiguous. In many cases the loss is attributed ad hoc to the person who happened to be on duty at the time of the event. In attributing the incidence to the trigger, instead of the situation, the system stakeholders typically become sloppy and careless about the design features that could have prevented the incidence (Harel, 2010, download).

The term "human error" often refers to an unintentional action that triggered a failure. Such definition is commonly used in studies of organizational behavior. The problem with this definition is that in many cases, the loss cannot be attributed to any unintentional action, or even to a judgment error. In these cases, this term should rather be attributed to the interactive complexity, namely, to operating the system in exceptional situations.

Another problem in using the term "human error" is that it may apply to various people roles, such as system developers, operators or accident investigators, and to various situations, such as system design, operation or marketing. The reported guide is about preventing incidences commonly attributed to errors that occur during the operation. To avoid confusion, the guide and this article use the term "operational errors". In this article, the term "error" may also be used as a shortcut for "operational error".

Resilience is a system property enabling safety assurance. According to a new definition "A system is resilient if it can adjust its functioning prior to, during, or following events (changes, disturbances, and opportunities), and thereby sustain required operations under both expected and unexpected conditions". This definition applies also to this guide. In this guide, expected conditions are those that follow known triggers, and unexpected conditions are those due to interactive complexity. The purpose of resilience engineering and architecting is to achieve full or partial recovery of a system following an encounter with a fault that disrupts the functionality of that system.

Zonnenshain & Harel (2013, download) argue that system failure is mostly due to operating in exceptional situations, because the operational procedures are specified with expectations about the operational context, which is often implicit. The authors defined predictable exceptional situations as those due to disturbances, namely, to predictable exceptional events. Examples of predictable exceptional situations include: under risk of an external hazard, extreme operational condition (such as slippery road), hardware failure, power failure, communication failure, and state mismatch, due to improper event (such as an operator's action), generated or received in a wrong scenario (for which a response procedure was not defined). Exceptional situations are the result of budget and delivery time constraints, and therefore are error-prone. The results of operating in exceptional situations are often unpredictable.

Unpredictable situations are due to missing or wrong specifications, to design mistakes, or to implementation errors (software bugs). Examples of unpredictable situations include: exceptional machine state (which is irrelevant to a particular stage in a particular operational procedure), and exceptional context (not mentioned in the system requirement specification document). Because incidences are often associated with unpredictable situations, operational resilience may be redefined as a measure of the system persistent to unpredictable situations. Root-cause analysis of many case studies reveals two key human-related sources of failure: latent hazards and delay in the recovery procedure.

A model of resilient operation proposed by Zonnenshain and Harel (2013, download) describes typical cycles of system operation involving handling exceptional situations. According to this model, system resilience comprises three main features: reliability, troubleshooting and recovery. This guide proposes an enhanced version of the model. According to the new model, resilient operation involves the following features: relying on reliability and robustness, to retain normal operation, and responding to faults by operating in an exceptional situation. Successful operation then may ending up in either immediate recovery (rebounding) or through a session of safe-mode operation (adapting). Failure in the operation is defined by loss.

Responding to a hazard is quite complicated and consequently is error-prone: operators are typically trained to run the system in normal situations. Often, they are not familiar with the exceptional situation, and they do not know how to recognize and identify the hazard. Yet, they need to capture and identify the hazard and the particular circumstances in no time, and they need to respond accurately immediately. Often, they are expected to know and follow predefined procedures, which they never had any opportunity to learn and practice beforehand.

It is commonly assumed that the system resilience relies on the capabilities of the operators and the organization. However, Zonnenshain and Harel (2013, download) pointed out that because the behavior of the human operators is unpredictable, system engineers typically try to do their best to automate as much as they can. However, the results might often be even more risky, when the operators are not aware of the details of the solution.

Occasionally, one of the system units may receive an exceptional event (a slip). As a result, the operational scenario needs to change. For example, in case of a unit failure, the operational scenario may change to "Unit Replacement". If all the system units operate now according to the new scenario, then the system is scenario compliant. Otherwise, if a subset of the system units still operated according to the pervious scenario, then the system reaches a state of internal inconsistency (Zonnenshain & Harel, 2009, download; Harel & Weiss, 2011, download).

The inconsistency can occur in between the machine's units. However, more frequent is the case of inconsistency due to poor coordination between the machine and its operators regarding the active scenario, for example, when the human operator is not aware of a change in the machine situation. Inconsistent scenario assumptions are perceived as unpredictable. Unless the machine provides the operator with a reset feature, enabling seamless resumption to normal operation, the inconsistent state might end up in an incidence (Zonnenshain and Harel, 2013, download).

The purpose of the guide is direct designers to assuring resilient operation proactively. The proactive strategy directs the designers to identify hazards before they materialize into incidences or accidents and taking necessary actions to reduce the safety risks (Weiler & Harel, 2011, download). Mishaps should not be regarded as force majeure (Harel & Weiss, 2011, download). Rather, system engineers should be able to design and develop systems that can operate safely even when in unusual operational situations. Proactive resilience assurance is about facilitating the system operation, including when in exceptional situations, ensuring safe behavior in such situations, and facilitating the recovery (Zonnenshain & Harel, 2013, download). The guide proposes guidelines for identifying and preventing design mistakes, such as error-prone operational procedures.

The guide proposes a hyper principle, which is about the designers' responsibility to prevent operational errors. It also proposes a set of sub principles, considering the unique properties of the human operators. Harel and Weiss (2011, download) proposed to formalize the system design (for resilience assurance) and suggested that "the system design may include:

- An interaction protocol, defining the rules to control the event processing and the state changes, according to the operating scenario
- A scenario tracker, which may hold and update a record of the operating scenario
- An event interpreter, which may verify that the events received comply with the operating scenario"

The guide proposes a way to implement the STAMP paradigm of self control, according to operational rules. The guidelines in this guide include instructions about designing a control unit in charge of handling the STAMP paradigm. The operational rules should be defined explicitly, and the system should constrain its operation according to these rules. Moreover, the guide recommends implementing the rules in dedicated control units, and specifying the system response in case of deviations from the constraints.

Additional components (sensors, indicators, algorithms, controls), required to implement the STAMP paradigm, are not only costly, but also risky, because they are liable to fail, providing opportunities for new kinds of incidences. Zonnenshain and Harel (2013, download) termed faults in safety add-ons as Secondary. The guide proposes guidelines for evaluating the new risks with those of the original configuration.

Common methods for risk assessment, based on estimates of its probability and expected damage, are not applicable to unexpected events, because the data required to get such estimates are unavailable. When dealing with potential events that did not materialize yet, we have no other choice but to rely on models of system failure. This method was demonstrated by Weiler and Harel (2011, download).

Following an incidence or an accident, the people involved typically focus on accountability issues rather than on improving the safety. In emotion-driven organizations, where the safety culture is biased by accountability, incidence investigations often obey the "blame and punish" script. Emotion-driven response to incidences prohibits improving resilience, because the investigations do not focus on the design changes needed to improving the resilience. On the other hand, when the organization adopts safety culture, the investigations include recommendations for design changes, and the management promotes implementing these recommendations. The guide proposes a procedure for continuous improvement of the system resilience by learning from mishaps, preventing this bias (Weiler & Harel, 2011, download).
The project focuses on achieving the following goals:

- Propose guidelines for preventing operational failures by design
- Propose a way for qualitative evaluation of design alternatives
- Propose means to trace the events preceding incidences, and a method for reporting and concluding about design changes that may prevent similar incidences.

The reported guide is based on a paradigm about the system vulnerability to use errors, namely, the Human Factors version of Murphy’s law (Harel, 2010, download):

If the system enables the users to fail, eventually they will!

This paradigm implies that it should be the developer’s responsibility to design the system such that use errors are impossible. Specifically, in order to facilitate the operators’ intervention, the machine should provide them with information about its state, and the information should be presented in forms considering the limitations of the human perception.

The guide

The guide proposes guidelines for Resilience-oriented Design (ROD) which focuses on the unusual situations. The guidelines concerns requirements and methods for alarming the operators about changes in the machine state, of which the operator must be aware, and about exceptional situations, for which the operator's intervention may be required. A preliminary guide for ROD was introduced by Zonnenshain & Harel (2013, download).

In the current version, the guidelines are arranged in two primary sections: proactive design, about preventing incidences, and reactive design, about learning from incidences.

The core of proactive design is a resilience model, describing five strategies for defending against hazards. The guide proposes defining firewalls and operational facilitators for coping with hazards. The firewalls are for disabling hazards and escalation. The facilitators are about enabling three operational activities: hazard detection, troubleshooting and recovery.

The guide is a descendent of project of developing a suite of software tools for usability testing (Harel, 1999, download). The current project is the outcome of a session of two meetings of the ILTAM/INCOSE-IL Risk Management Working Group in 2010, in which we discussed various risks of operational errors. By the end of the second meeting, we came up with a preliminary guide, classifying errors in six categories, and proposing means to prevent them (Zonnenshain and Harel, 2013, download). One of these categories was about the user awareness about risky situations. The guidelines developed for this kind of errors were implemented in a case study about the effectiveness of medical alarms designed according to IEC 60601-1-8. The conclusions were sent as comments to the standard working group (Harel, 2011, download).

Prior to this project we had two pilot projects, about the risks of unexpected events (Harel & Weiss, 2011, download) and about managing the risks of driving errors (Weiler and Harel, 2011, download). The current project started in 2012, with the aim of developing three deliverables:

- A model of resilient operation, describing the ways systems typically behave in exceptional operational situations
- A guide for avoiding failure modes described using the model.
- A database of case studies, to validate and evaluate the effectiveness of the guide.

Conclusions

The output of this project is a guide for system engineers, about how to avoid common design mistakes which often result in operational errors. The guide proposes a way to integrate the guidelines in common procedures for system development, according to the traditional practices and routines.

As far as we know, this is the first guide ever proposed on this topic. We believe that system engineers will have to consider resilience in the system design, and that this guide will show them how. We believe that this guide will be accepted as a common methodology for resilience assurance, will evolve further to a basic discipline, taught in schools for system engineering.

The guide was validated using a variety of use errors, exemplified in case studies. The plan was to create a database of mishaps, and to ask system engineers of various backgrounds to contribute to the database from their own experience, and to help with the validation by providing feedback about the effectiveness of the guidelines. The validation methods employed were by peer review, based on the ILTAM/INCOSE_IL working group, and by evaluating the benefits of applying the guidelines to each of the case studies in the database of 67 cases, and presenting statistics of these evaluations. The validation process has led to some conclusions about changes required in the content and the format of the model and the guide. The guide was announced last year at the INCOSE Annual International Symposium (download).

The current version covers many important issues in resilience assurance, yet we already have a long list of issues that we intend to include in subsequent versions.

The current version of the guide is interactive, and is available on-line (explore the guide). Our vision is that system engineers will contribute to the body of knowledge, by sharing their experience using the guidelines, by commenting, and by adding their own recommendations.

References

Harel, A., 1999. "Automatic Operation Logging and Usability Validation" *Proceedings of HCI International '99*, Munich, Germany, Vol. 1, pp. 1128-1133 (download).

———, 2010. Whose Error is This? Standards for Preventing Use Errors, *The 16th Conference of Industrial and Management Engineering, Tel-Aviv, Israel*. (download)

———, 2011. "Comments on IEC 60601-1-8". *Letter submitted to IEC/TC 62 working group.* (download)

Harel, A. & Weiss, M., 2011. "Mitigating the Risks of Unexpected Events by Systems Engineering". Paper presented at The Sixth Conference of INCOSE-IL, Hertzelia, Israel (download)

Weiler, M. & Harel, A., 2011. "Managing the Risks of Use Errors: The ITS Warning Systems Case Study". Paper presented at The Sixth Conference of INCOSE-IL, Hertzelia, Israel. (download)

Zonnenshain, A. and Harel, A., 2009. "Task-oriented System Engineering". Paper presented at the INCOSE International Symposium, Singapore. (download)

———, 2013. "Resilience-oriented design". Paper presented at The Seventh Conference of INCOSE-IL, Hertzelia, Israel (download).

———, 2013a, "Towards families of resilient systems". Paper presented at The Yossi Levin Conference, Technion, Haifa, Jan. 9th, (download)

———, 2015, A practical guide to assuring the system resilience to operational errors (A. Zonnenshain, A. Harel). INCOSE Annual International Symposium, Seattle (download).