Resilience-oriented Design

Dr. Avigdor Zonnenshain
Rafael

Avi Harel
Ergolight

Abstract. The article introduces a guide for designing resilient systems. The Resilience Assurance Guide (RAG) is based on the System Resilience Model (SRM), a model describing activities typically involved in incident management. The SRM integrates proactive and reactive disciplines of resilience assurance, and the RAG provides corresponding instructions for tackling threats and for learning from incidents. The article uses a case study, of the AF447 accident, demonstrating how the RAG can help mitigate operational risks.

PRIOR STUDIES

System resilience. The resilience of a **system**[1] is defined here as a measure of the system ability to avoid incidents. An **incident** is defined here as a situation of potential loss (such as casualties or financial loss) during its operation. Another definition of this term is in the home page of the INCOSE resilient systems working group (RSWG)[2]. A **mishap** is an incident in which the loss is materialized.

System resilience is a key factor in assuring important operational features, such as safety and productivity, as well as in reducing operational costs, such as installation and maintenance expenses. Consequently, system resilience is also a key factor in creating a positive user experience.

Resilience Engineering. Hollnagel, Paries and Woods (2011) defines Resilience Engineering as the ability:

[1] The term System denotes here an Extended System, referring to a machine combined with its operators.
[2] http://www.incose.org/practice/techactivities/wg/rswg/

a) to respond to what happens,

b) to monitor critical developments,

c) to anticipate future threats and opportunities, and

d) to learn from past experience - successes as well as failures.

Whereas conventional risk management approaches emphasize calculation of failure costs and probabilities, Resilience Engineering is about designing robust operational procedures, in the face of ongoing production and marketing pressures[3].

Sources of operational failures. People, including designers and many accident investigators, often confuse between the sources of the incident and the event that triggered the incident (Dekker, 2006). A common bias in handling accidents is by hindsight, attributing them to the human operators or users (the "bad apple"), who theoretically, at the time of the incident, could prevent the incident, but did not (Dekker, 2007). Accordingly, most of the accidents are typically attributed to human errors[4].

In attributing the incident to the trigger, instead of the situation, the system stakeholders typically become sloppy and careless about the design that could prevent the incident (Harel, 2010). The new approach to system failure is that incidents and normal operation are two perspectives of the same behavior. The actual perspective is related to the actual point of the Efficiency-Thoroughness Trade-Off (ETTO). According to the new approach, incidents are better described as extreme deviations from the normal operational procedures[5].

Incidents are often the result of operating the system in an exceptional situation, for which the design was incomplete, and in which the system was not tested properly (Roberts, Isensee & Mullaly, 1998). As a result, the machine's[6] or the operators' behavior are sometimes perceived as unexpected, and the subsequent response by the other party is perceived as unpredictable. Therefore, the same activity that works fine in normal operation might result in an incident, when the situation is exceptional. This is typically the case of state mismatch, when the operators assume a wrong context (Zonnenshain & Harel, 2009).

Resilience assurance. Although, as the term Resilience Engineering suggests, the intention is to develop resilient systems, so far, most of the studies about resilience engineering focus on failure analysis. Common practices of system requirements specification and design propose very general guidance with very few instructions for assuring the system resilience.

Until recently, there were no practical guides available, which provide detailed instructions for system engineers about how to incorporate human factors in the design for resilience. Consequently, system resilience still relies on the talent and skills of system engineers and experts in the application domain.

A pioneering approach to resilience assurance is described in the STAMP model, by Leveson (2004). This model may help system designers consider important safety issues, and guide them in preventing incidents.

---

[3] E. Hollnagel web site: https://sites.google.com/site/erikhollnagel2/resilienceengineering

[4] http://www.reliability.com/healthcare/articleshcp/jan_08_Cost%20and%20Truths%20of%20Human%20Error.pdf

[5] Hollnagel, Erik, (2004) https://sites.google.com/site/erikhollnagel2/fram

[6] Part of the instructions for operational resilience is applicable also to static equipment, such as buildings and installations.

Recently, a pilot resilience guide was developed by the Iltam workgroup for risk management. The guide was based on a simple resilience model, comprising six failure modes, and it proposed few guidelines for mitigating the risks associated with these failure modes. The guidelines were evaluated using a small set of 15 mishaps. Of these, eight were of "celebrated" accidents found on the internet, and seven were proposed by members of the workgroup.

Recent studies. Recently, The Gordon Center for System Engineering at the Technion, Haifa, in collaboration with INCOSE-IL and Iltam, has conducted new studies about ways to elevate the operational resilience, by incorporating human factors in the system design, using models of system failure. We have reported about two such studies, about managing unexpected events (Harel & Weiss, 2011) and operational risk management (Weiler & Harel, 2011).

A Guide for Resilience-oriented Design. In an ongoing study, reported in this article, we have developed a model of system resilience (the SRM), and a guide for resilience assurance (the RAG), which is based on the SRM. The SRM describes incidents in terms of operational and learning activities, considering the limitations of users and operators in coping with exceptional situations. The pilot RAG provides instructions and guidelines for specifying and designing resilient operational procedures, and for providing means to facilitate learning from incidents and mishaps.

Development of the resilience guide. The pilot resilience guide, developed by the Iltam workgroup for risk management, had only few guidelines. These guidelines were evaluated using a small set of 15 mishaps, based on a simple resilience model. Since then, the resilience model and guide are being developed gradually, by analysis of additional mishaps. Occasionally, the new analyses revealed new failure modes. Following the analyses, we occasionally improved the guide by adding new guidelines that hypothetically could prevent the new failure modes.

Validating the Guide. Our way to validate the guide is by case studies, selected from several sources:

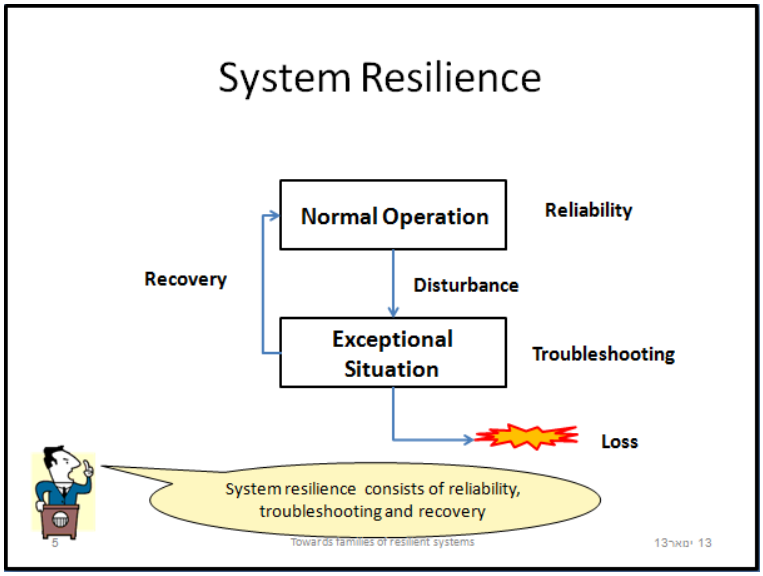
- Published analyses of mishaps
- Published stories about well-know accident
- A database of incidents maintained by Iltam.

The procedure to conduct a case study is by:

1. Create a list of all factors that can be regarded as sources for the failure
2. For each factor, check if the guide could mitigate it, should the designers used this guide at the development stage
3. Heuristically evaluate the effectiveness of the guide over all the factors.

THE SRM

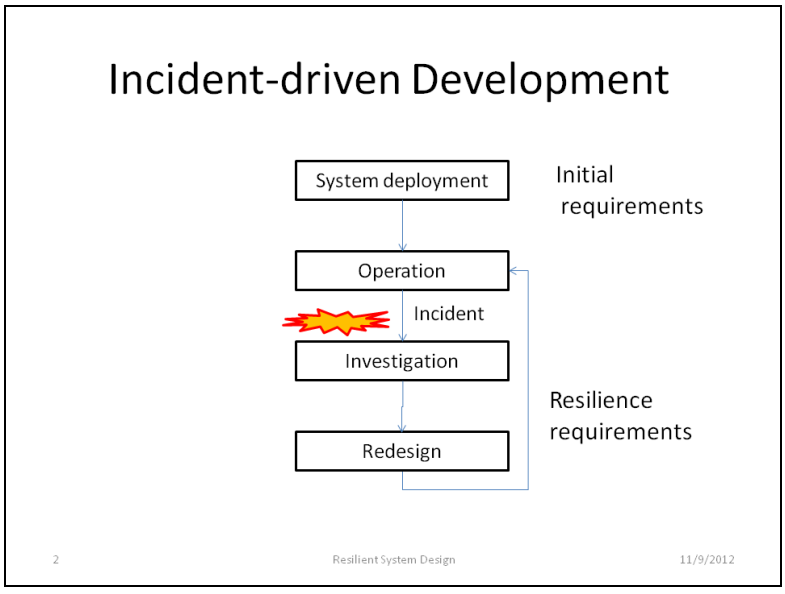
The system resilience is defined as an emergent property, associated with three system properties: reliability, troubleshooting and recovery. The following chart depicts this definition:



Incident-driven development. The resilience model assumes that resilience development is gradual, through cycles triggered by incidents. Each incident is followed by two operational activities:

- **Investigation**, including capturing and identifying the incident, and concluding about steps and means required to avoid similar incidents
- **Redesign**, including changing the requirement specifications, designing and implementation.

This is depicted in the following figure:



Disturbances. An event that triggers an incident is called a disturbance. Types of disturbances include:

- An external event, such as an obstacle on a road, or an enemy boat detected by radar
- A hardware unit or component failure

- Power failure, such as due to battery change or weak connection
- Communication failure
- State transition due to a software bug
- Inadvertent activation of a control or a feature
- Exceptional changes in production rate.

Responding to a disturbance. If the exceptional situation is predictable at design time, then the design may include safety add-ons constraining the system to operate according to rules. A safety add-on may include:

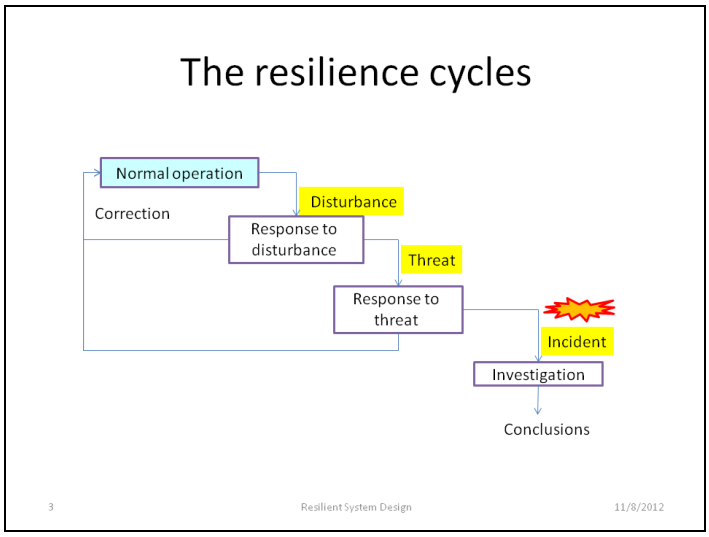
1. Special, dedicated sensors,
2. Special rules or algorithms enabling to detect deviations from the normal behavior,
3. Rules about how to resume normal operation, and
4. Special safety controls or utilities.

Threat generation. Ideally, the system may recover from a disturbance easily and resume normal operation instantly, with minimal attention and effort of the operator. Practically, automatic disturbance resolution is rare. More common is the case that the system needs to sustain its normal operation, until the operators find the way to fix the problem. When this is the case, the disturbance transforms into a threat.

The resilience cycles. The top layer of the model defines two resilience cycles:

- A short cycle is an operational cycle of responding to a disturbance
- A long cycle is an operational cycle or responding and recovering from a threat.

The resilience cycles are depicted in the following chart:



The operation cycle consists of departures from normal operation due to disturbances, which may be resolved instantly, either automatically or manually. If a disturbance is not resolved instantly, it transforms into a threat, which requires allocation of human attention and

intervention. If the operator fails to resolve the threat, or to resume normal operation, then the threat transforms into an incident, which breaks the normal operation cycle.

The operational context. Systems are designed to operate according to scenarios. This means that the response of any unit to any event is designed based on assumptions about an operating scenario. In normal operation, all system units assume the same scenario. This scenario is called the operational context.

Exceptional situations. System design and testing is always constrained by budget and schedule. To work around these constrains, typically, the development activities are prioritized, so that the initial focus is on the procedures that implement the primary functions. Because delivery time is always a constraint, the exceptional situations are typically error-prone, and the results of operating in exceptional situations are often unpredictable.

An exceptional situation may be classified as either predictable or unpredictable, as follows:

Predictable exceptional situations. Predictable exceptional situations are those due to disturbances, namely, to predictable exceptional events. Examples of predictable exceptional situations include:

- Under risk of an external threat
- Extreme operational condition (such as slippery road)
- Hardware failure
- Power failure
- Communication failure
- 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).

Escalation. Occasionally, one of the system units may receive an exceptional event (a slip). In response, the operational scenario may 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 context compliant. Otherwise, if not all the system units comply with same context, then the system reaches a state of context inconsistency (Harel & Weiss, 2011).

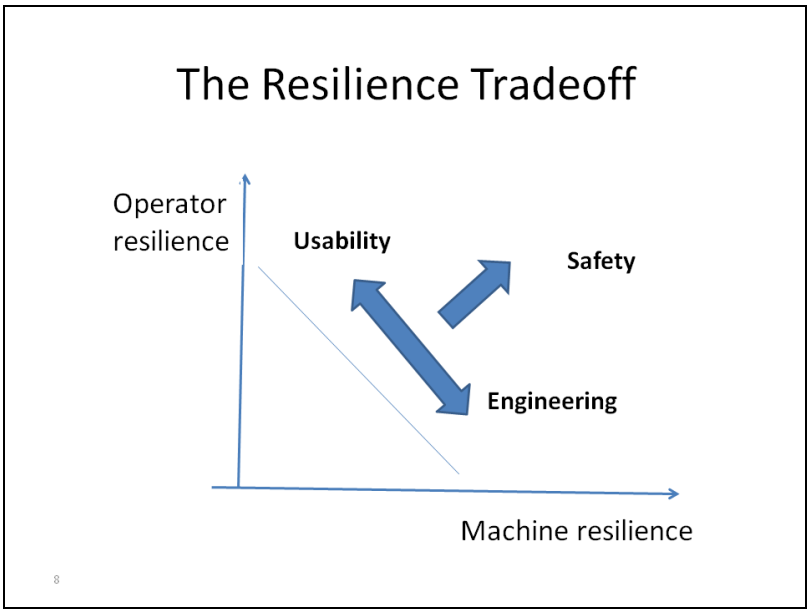
Unpredictable situations. Unpredictable situations are due to missing or wrong specifications, to design mistakes, or to implementation errors (software bugs). Examples of unpredictable situations include:

- Exceptional machine state (which is irrelevant to a particular stage in a particular operational procedure)
- Exceptional context (not mentioned in the system requirement specification document)

Because incidents are often associated with unpredictable situations, operational resilience may be redefined as a measure of the system persistent to unpredictable situations.

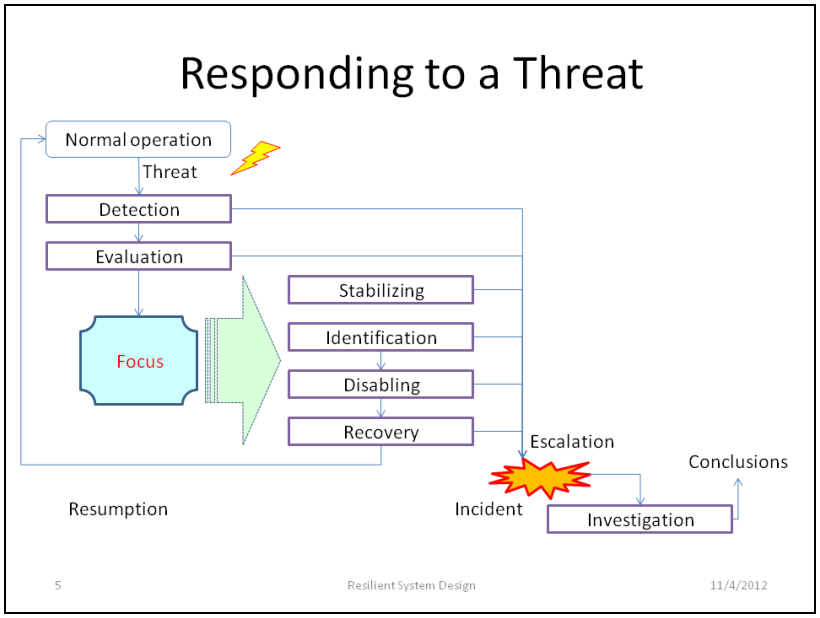
Secondary faults. The additional components required to implement the controller (sensors, algorithms, controls) are not only costly, but also risky, because they are liable to fail, providing opportunities for new kinds of incidents.

Task allocation. Because the behavior of the human operators are unpredictable, system engineers do their best to automate as much as they can. However, the more reliable the automation, the less the human operator have opportunities to learn how to handle exceptions (Bainsbridge, 1983). For example, inadequate crew knowledge of automated aviation systems featured as a factor in more than 40 per cent of accidents between 2001-2009[7]. The following chart depicts the human-machine resilience tradeoff chart:



Handling threats. Threats are controlled by the human operators, assisted by the machine. Threat recovery is an interactive activity, in which the machine informs the operators about the situation, the operators integrate this information with their own information and knowledge, and act to overcome the threat. This kind of interaction is depicted in the following chart:

[7] http://www.airtrafficmanagement.net/2012/07/analysis-tales-of-the-unexpected/



The activities involved in handling these interactions consume attention and require intervention by the human operator. Because the operators' attention is required also to other problem solving activities, these interactions are error-prone.

Escalation. If any of the interactions described above fail, the system might enter an ambiguous or inconsistent state. The inconsistency can occur in between the machine's units, as was the case with the Therac 25 accidents, or between two sub systems, as was the case with the friendly-fire accident in Afghanistan, 2001. However, more frequent is the case of the inconsistency in which the human operator is not aware of a change in the machine situation.

Inconsistent states are perceived as unpredictable, and the operators typically fail in identifying the source for the ambiguity. Unless the machine provides the operator with a reset feature, enabling seamless resumption to normal operation, the inconsistent state might end up in an incident.

Incident investigation. Often, when the organizational culture is around survival, incident investigation is emotional-driven, following the "blame and punish" paradigm. Emotion-driven response to incidents prohibits improving resilience, because these investigations are careless about 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.

Special tools for incident reporting and information sharing may facilitate the shift from survival culture to safety culture.

THE RAG

Human factors. The ideal way to handle disturbances such as unintentional actions is by automation. However, a main concern of the SRM is about situations such as hardware failures, in which a human intervention is required to identify and recover from the

disturbances in time. Accordingly, the SRM is based on a paradigm about the system vulnerability to use errors, namely, the Human Factors version of Murphy's law (Harel, 2010):

> 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. Accordingly, the SRM focuses on 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.

The guide assumes that resilience relies on the capabilities of the operators and the organization. The designer's task is to facilitate the operational procedures, as well as the procedures for assuring safety climate.

Proactive resilience assurance. Proactive resilience assurance is about ways to reduce the probability of operational and learning activities, considering the limitations of users and operators in coping with exceptional events, to design safe responses and to facilitate seamless recovery.

The RAG may help the designers to specify means to cope with disturbances and threats. Ways to handle predictable situations include prevention (if possible), reducing the possible failure modes subsequent to the threat, and minimizing the damage

The resilience-oriented requirement specification should include explicit instructions and guidelines for preventing all possible operational failures. A general requirement is that the system should handle all exceptional situations, as detailed in the resilience model.

The paradox of the resilience tradeoff. The operators' response to an exceptional situation is that which they applied successfully in the past, in normal situations. This implies that in abnormal situation due to a threat, we should seek an automated solution.

The limitations of automation. Two main limitations should be considered:

1. **System Complexity**. Automatic recovery means increased machine complexity, due to adding sensors, rules and special algorithms to enable automatic recovery. And, the added components are also error-prone, providing more opportunity for system failure
2. **Irony of automation**. The more reliable the machine, the less is the operator competent to solve problems. Because the human operators have less opportunity to experience the exceptional situation, they do not know what to do when they face it (Bainsbridge,1983).

Facilitating the operator's part. It is evident from the history of accidents that the human operators sometimes fail to follow the operational procedures. Therefore, the operational resilience depends on the way human factors are incorporated in the operational procedures, in normal and in exceptional situations.

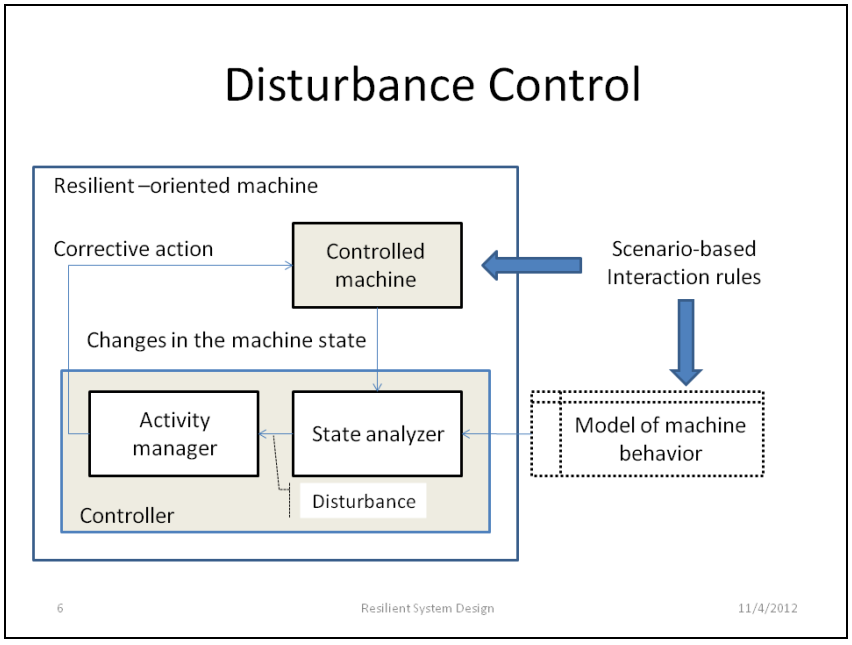
The design should assume that both the original Murphy's Law and its Human Factors variant apply to the system. The designer's duty is to make sure that the machine will resist human errors. In the design we should assume that the operators cannot follow the exact machine state, and hence they might activate a feature which is irrelevant to the machine state.

The system design should include guidance about the system states, and should prevent activating features which are irrelevant to the machine state in process.

Enforcing the organization's part. The system behavior is often affected by organizational considerations, such as a policy about allocating authorities and responsibilities between the organization and the operators. For example, the organization may constrain the setting of alarm thresholds to a certain range, allowing the operator to set other thresholds within these constraints.

The design should reflect the organizational considerations, enabling organizational customizing, and operator's customizing subject to the organizational constraints.

Handling disturbances. Ideally, we would like to avoid threats, by automatic disturbance resolution. The system can constrain its own behavior by a safety add-on. The additional controller needs to implement a model of normal system behavior, to identify exceptions and to provide corrective actions. This behavior is depicted in the following chart:



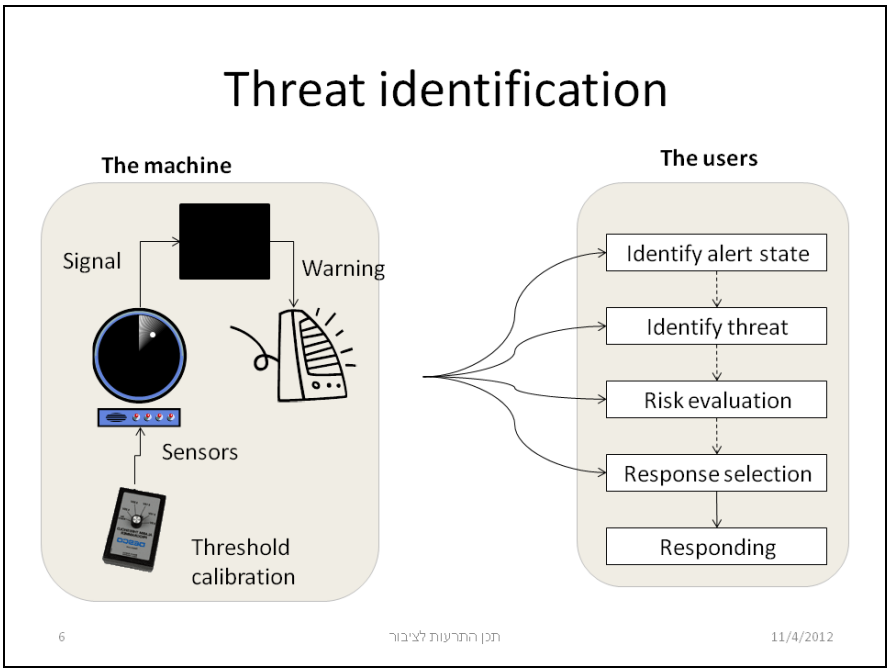
Handling unpredictable situations. Automatic disturbance resolution is only applicable to situations which are predictable at design time. It is the role of the human operator to handle unpredictable situations which might follow a failure in any of the recovery activities mentioned above.

Facilitating threat control. The following principles are applicable to threat control:

- Maintenance of a common situational database
- Using warnings to synchronize the users with the machine state

- Ongoing verification of the situation compliance with the requirement specifications

Facilitating the operator's tasks. The following chart depicts the activities involved in threat identification.



The right hand side presents the operator's mental activities in responding to a warning signal. The machine, on the left side, should support the operator's mental activities, by encoding the warning signals properly. Failure to support any of the user's tasks should result in an incident.

Ideally, the warning should evoke an immediate operator's response, which will enable to fix the problem. However, this is not possible because of the paradox of automation; if there is only one proper response then this means that the machine can handle it automatically. On the other hand, if the operators are used to respond in one way to a particular warning, they repeat this response also when the problem is different, and requires a different response.

Facilitating situation awareness. The following list is of guidelines for ensuring that the users are aware of the exceptional situation:

- Ongoing verification of the warnings audibility
- Validating the detection of threats
- Validating the generation of warnings in case of threat detection
- Resilience to third-degree threats, such as those due to nuisance avoidance
- Rule based detection of unpredictable situations
- Design for the user's attention to warning signals
- Validating the user's becoming alert by the warning, based on the principle of natural alerts

- Design for the operator′s vigilance
- Design for detecting second degree threats

Facilitating risk evaluation. The following list is of guidelines for ensuring that the users of the warnings can evaluate the risk properly:

- Assuring the operator′s awareness of the machine situation
- Facilitating the operator′s decision making
- Assuring proper estimation of the risk
- Principle of prior warning
- Attracting the user′s attention to the exceptional situation
- Allocating operator′s sensory channels to threats
- Assuring the distinctiveness of warning signals
- Principle of reflexive response
- Standards for allocating warning patterns
- Reduction of competing alarms
- The principle of constant alarm characteristics
- The use of training modes

Mitigating system stabilization. The following list is of guidelines for ensuring that the operators can stabilize the machine state:

- Stabilizing by freezing
- Stabilizing by fading out
- Manual stabilizing
- Assuring inter unit coordination

Assuring threat identification. The following list is of guidelines for ensuring that the operators can identify the threat:

- Guiding the operators about the need to intervene
- Guiding the operators about the required activities
- Adapting the means to the source of threat

Concealing the threat. The guidelines for concealing the threat depend on the source of threat

Assuring resumption to normal. The following list is of guidelines for facilitation the resumption from the exceptional situation:

- Defining reset points and rescue points in the operational procedures, intended to use when rolling back
- Guiding the operators about rolling back and resumption
- Specific resumption operational procedures
- Recovery time frames.

Assuring learning from incidents. The guide is based on the iterative approach for gradual resilience development, presenting instructions for detecting, reporting, retrieving and investigating incidents.

Special engineering tools, such as for logging of the system behavior, for incident identification and reporting, enable to provide information required for learning from incidents and mishaps.

The following list is of guidelines for facilitation the procedures of learning from risky events and mishaps:

- Collecting all relevant data, including disturbance, threat, failure, people's reports
- Assuring safety climate, highlighting the management's responsibilities
- Assigning the investigator
- Analysis of alternative fixes.

Responding to external threats. The following list is of guidelines for additional means for resilience assurance, specific to external threats:

- Means and procedures to set and control threshold values for measurements in normal operation and in risky situations
- Characteristics of sound and visual advance warnings.

Responding to hardware failure. The following list is of guidelines for additional means for resilience assurance, specific to hardware failures:

- Redundancy of critical units
- Detecting hardware failures
    - The use of sensors to detect first degree failures
    - Detecting failures by constrain verification
    - Detecting secondary failures by time constrains
    - Redundancy in the warning system
- Stabilizing the system following hardware failure
    - Emergency shut down

    - Indications about the machine state and the operation state
    - Warnings about inconsistencies during stabilization

- Troubleshooting
    - Direct mapping from intention to action
    - Training for troubleshooting

- Recovery
    - Configuration conservation
    - Designing default values

Responding to an unexpected event. The following list is of guidelines for additional means for resilience assurance, specific to unexpected events:

- Rule-based threat detection

- Means to facilitate learning from incidents, including:
    - Black box recording of all activities
    - Recording of indicators of detected threats
    - Automatic "think aloud" enquiries to the operators about problems

Responding to use errors. The following list is of guidelines for additional means for resilience assurance, specific to use errors:

- Scenario-based design

- Means specific to initial operation, including:
    - Direct access to information
    - Direct transition from intention to action

- Means specific to daily operation, including:
    - Function-based control aggregation
    - Direct mapping from task to action
    - Means for assuring operator situation awareness
    - Enforcing the operator to select a safe option

- Means to facilitate the transition from a novice to a skilled operator, including:
    - The principle of seamless transition
    - The principle of consistent response to user actions
    - The principle of consistent control location

CASE STUDY

The AF 447 accident. This section demonstrates a validation cycle based on a single case study, the accident of Airbus A330-200 of AF 447[8] .

Air France Flight 447 (abbreviated AF447) was a scheduled commercial flight from Rio de Janeiro, Brazil to Paris, France. On 1 June 2009, the Airbus A330-200 airliner serving the flight crashed into the Atlantic Ocean, killing all 216 passengers and 12 aircrew.

The main reason for the accident was that he less experienced co-pilot took the wrong action, pulling the stick all the way back, probably, according to the wrong prediction that this will enable the airplane to bypass the storm. In high altitude, raising the angle-of-attack typically results in stall.

Accident analysis. The aircraft crashed following an aerodynamic stall caused by inconsistent airspeed sensor readings, the disengagement of the autopilot, and the pilot making nose-up inputs despite stall warnings, causing a fatal loss of airspeed and a sharp descent. That led to the craft dropping 38,000 feet into the sea in four minutes. The analysis of the accident of this case study is described here by applying the SRM.

The result was a series of moves that reduced the plane's speed and placed it in a nose-up position causing an aerodynamic stall.

Situation analysis. Prior to the erroneous pilot behavior, other events made the situation exceptional:

- The airplane, heavy with fuel and ice, could not climb to escape the storm
- The captain left the cabin, nominating the less experienced co-pilot in charge
- Seeking to avert a zone of severe turbulence the co-pilots disengaged the autopilot and took manual flight control, in which they were not trained.
- The other co-pilot fixed the setting of the radar system, which was not set properly in the beginning, yet resulted in worsening the co-pilots stress situation
- Inconsistent airspeed sensor readings due to blockage of Pitot tubes by ice

Analysis of the task allocation. This action exemplifies the following principles presented in the SRM:

- The role of complexity due to adding safety features: the co-pilots who were used to fly airplanes in high altitudes in automatic mode, did not know how to fly them in manual mode
- The Human Factors variant of Murphy's Law: if the machine enables the human operator to fail, eventually, they will
- The irony of automation: when the human operators are unfamiliar with the situation, such as due to improper automation or to automation break, often they do not know what to do

[8] http://en.wikipedia.org/wiki/Air_France_Flight_447

- The paradox of the resilience tradeoff: the behavior that controls the situation is that to which the human operators are used.

Threat identification. Although the stalling alarm was clear and loud, and it sounded 75 times, the co-pilot in charge kept pulling the stick back, continuously raising the angle-of-attack, disregarding the alarm. From the recording it is clear that the co-pilots understood that they lose control but they did not attribute the exceptional situation to the way they used the sticks.

The display in the cockpit did not include all the information required to convince the co-pilot in charge that he needs to regard the Stall alarm, and to reduce the angle-of-attack. Because the sticks of the two co-pilots were asynchronous, the other co-pilot was not aware of the fact that the co-pilot in charge was pulling the stick all the way up.

Recovery. The recovery was not successful, because the co-pilots did not identify the stall situation, and were not aware of their contribution to the situation. Probably, the co-pilots never heard the stall alarm before. Obviously, they were not trained to handle stalling situations.

Learning from the mishap. The final accident report includes various recommendations, primarily about enforcing pilots to train operation in Manual modes.

Hypothetical resilience assurance. By applying the RAG to this case study, hypothetically, the design could have mitigated some of the sources for the accident. Such conclusion may be encouraging, because according the Swiss Cheese model[9], it is sufficient that we mitigate only few of the risk sources.

Avoiding entering the thunderstorm. The RAG recommends that prior to taking an action, the machine may present a review of the outcome. In the case study, the preview could be momentary, indicating a potential stall, or delayed, indicating the effectiveness of climbing within the non-stalling constrain.

Apparently, the cockpit computer did provide the momentary preview, in the form of Stall alarm. However, because the investigation reports do not mention a preview of the delayed preview, it is most probable that the co-pilot who pulled back the stick was not aware of the ineffectiveness and risks of his action.

Authority of crew members. This issue is beyond the scope of this RAG, but it should be targeted in subsequent guides.

Changing to manual control. Apparently, the co-pilots were experienced in manual control in takeoff and landing, but not in maneuvering the airplane in high altitudes. The RAG recommends special training of situations in which the operators are not familiar, and particularly in troubleshooting. From the investigation reports it is clear that training is considered a key factor in enabling the accident.

Wrong setting of the radar system. We do not have enough information that may enable us to analyze why the setting was incorrect, and how the co-pilots realized that this was the case,

[9] http://en.wikipedia.org/wiki/Swiss_cheese_model

how they could be aware of it earlier. Therefore, we cannot see how the RAG could help with this issue.

Avoiding the risky pulling back of the stick. The RAG recommends providing the operators with preview information, namely, estimates about the probable results of maintaining the current operational situation. Such information could help the co-pilots realize that they should maintain the horizontal angle-of-attack.

Assuring threat identification. The RAG warns that in stressful situations people often fail to perceive correctly even very clear alarms. The current version of the RAG does not include methods to overcome this barrier.

At one point the co-pilot in charge told his colleagues that he was pulling the stick back continuously. The captain (who was back) instructed him to push it down, but this was too late. This part of the recording demonstrates the importance of communicating the thoughts of the people involved in the troubleshooting to other people. An appropriate guideline, proposing to adopt the "think aloud" technique used in regular usability testing, may be added to the next version of the RAG.

Operating in Manual mode. The RAG recommends that the operators are trained to solve problems. The air crew may have been trained to fly the airplane in manual mode in low elevations, but not in high elevations. The RAG proposes that training for troubleshooting may be possible by simulation of various kinds of disturbances in various exceptional situations, so that the operators have opportunities to experience troubleshooting in` such situations.

Synchronizing the crew members. The RAG recommends ensuring that all the operators can see all the updated information. The asynchronous operation of the two sticks prevented one co-pilot from realizing what the other co-pilot was doing. Obviously, designing according to this recommendation could hypothetically have prevented the accident.

Enforcing pilot training. This issue is beyond the scope of this version of the RAG, but it may be targeted in subsequent versions.

CONCLUSIONS

Clearly, the SRM highlights the main failure modes manifested in the case study. Obviously, the guidelines proposed in the RAG can help mitigate the major failure modes presented here.

It might seem too tedious to follow all the guidelines described in the RAG. For example, extensive training for flying an airplane in exceptional situations might be very costly, making it impractical. Following this kind of argument, it might be concluded that it is more practical to save the expenses until there is real evidence about the problem. The AF 447 accident provides such evidence, and this evidence is manifested in the formal accident report, which recommends on extending the pilots training.

The common practice of disregarding a failure mode because the known solution is too costly is unacceptable. The main conclusion of the case study may be that we should not wait for the mishaps. The Human Factors version of Murphy's law should be examined whenever a human operation is required. In the design we should always think of the option that the human

operator will fail to understand what is going on, and will behave illogically. In the design of a safety-critical system, we should find an affordable solution for each of the failure modes.

A practical way to implementation of the RAG may be through the following stages:

1. Identify the predictable exceptional situations
2. Prioritize the disturbances in the predictable situations by their risks
3. For each of the disturbances work out a solution with an affordable price
4. Implement.

Demonstration of this procedure may be the next step in the study of resilience-oriented design.

REFERENCES

Bainbridge, L. Increasing levels of automation can increase, rather than decrease, the problems of supporting the human operator. *Automatica, 19*, 775-779, 1983. Reprinted in: (1987) Rasmussen, J., Duncan, K. and Leplat, J. (eds.) *New Technology and Human Error, Wiley, Chichester*, pp. 276-283,

Dekker, S. *The Field Guide to Understanding Human Errors*, Ashgate, 2006

Dekker, S. *Just Culture: Balancing Safety and Accountability*, Ashgate, 2007

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

Harel, A. & Weiss, M. Mitigating the Risks of Unexpected Events by Systems Engineering, *The Sixth Conference of INCOSE-IL, Hertzelia, Israel*, 2011

Hollnagel, E. *The ETTO Principle: Efficiency-Thoroughness Trade-Off. Why Things That Go Right Sometimes Go Wrong*. Ashgate, 2009

Hollnagel, Paries and Woods: Resilience Engineering in Practice: A Guidebook, Ashgate, 2011

Leveson, N.G. A new accident model for engineering safer systems. *Safety Science 42(4)* 2004

Roberts, D., Isensee, S. and Mullaly, J. *Designing for the User with Ovid: Object-Oriented User Interface Development.* Macmillan Technical Pub, 1998

Treat, J. R., Tumbas, N. S., McDonald, S. T., Shinar, D., Hume, R. D., Mayer, R. E., Stanisfer, R. L. and Castellan, N. J. Tri-level study of the causes of traffic accidents. *Report No. DOT-HS-034-3-535-77 (TAC)*, 1977

Weiler, M. & Harel, A. Managing the Risks of Use Errors: The ITS Warning Systems Case Study. *The Sixth Conference of INCOSE-IL, Hertzelia, Israel*, 2011

Zonnenshain, A. & Harel, A. Task-oriented System Engineering, *INCOSE International Symposium, Singapore*, 2009

BIOGRAPHY

Dr. Avigdor Zonnenshain

Avi Harel

- President & CEO of ErgoLight institute for assuring operational reliability
- Chair of the Technical Committee (TC) for Usability of the Israeli Institute of Standards (SII)
- Board member of the Israeli branch of the Human Factors and Ergonomics Society (IHFES)
- Editor of the informal website for Israeli usability standards
- Chair of the SII workgroups for assuring the usability of alarms in medical equipment, public alarms and alarms in the process industry.