July 23, 2022

Essentials of Model-based Integration of Socio-technical Systems

A white paper by Avi Harel, Ergolight – ergolight@gmail.com

Top-level model of an STS

An STS may consist of human elements (operators, users, and stakeholders), subsystems that provide services to the human elements, and processes used to provide the services. The subsystems may include automated devices, engineered systems, and sub socio-technical systems. Part of the system units, notably the human elements, may be regarded as OEM black boxes.

The performance envelope

Typically, the value of an STS is the expected operational utility, defined as the optimal performance, constrained by the performance boundaries. In normal operation, the performance values should be in the performance envelope. Operation beyond the envelope is often costly, resulting in degradation of productivity and usability, and sometimes, in accidents (cf AF 296, 1988). System situations corresponding to operating beyond the boundaries are called exceptions.

Behavioral integration

The integration task is more than the traditional assembly, verification, and validation: it is about designing the coordination between the system elements and validating the system behavior proactively, at design time.

From black swans to engineering

The famous Murphy's Law is a conclusion from observations of system failure: if the operation might fail, eventually, it will. A conclusion based on the Black Swan Theory is that operators can prevent some of the failures, but not all of them.

Failure is due to operating in exceptional situations. A proactive version of Murph's Law is about responsibility: failure should be prevented by design. The design should disable potential ways of approaching the boundaries and should rebound from unexpected reaching the boundaries.

Behavioral complexity

Exception handling is by orders of magnitude more complex than the design for normal operation. Common practices optimized for normal operation are not adequate for designing for exceptions. They are extremely costly, and they do not provide the desired level of protection from hazards.

Learning from rare events

Proactive validation must be based on knowledge about the risks of exceptions. Unfortunately, exceptions are rare events, and the risks are unknown at design time. A way to cope with the barrier of rare events is by cross-industry sharing of protection methods. The challenge is to develop a general, universal model of exception handling, which may be used to customize the behavior of the system units in exceptions. For example, we can learn from problems in operating home devices, such as mode errors due to unintentional device settings. We can apply this knowledge to protect safety-critical devices from mode errors due to unintentional changes in the device settings.

The human element

A commonly accepted model of the human element, proposed by Card, Moran & Newell (1983) is in terms of Goals, Objects, Methods, and Selection rules (GOMS). Kahneman proposed that the mental processing involved in decision-making consists of parallel processing of two mental systems, which he called System-1 and System-2. System-1 is in charge of instant, reflexive reacting, and System-2 is in charge of thoughtful, rational thinking. The theory of System-2 applies to designing normal processes, and the theory of System-1 applies to testing the behavior in exceptional situations.

The HF version of Murphy's Law

The model describing System-1 assumes that humans are not perfect in doing what they intend to do. They often slip (cf Torrey Canyon, 1967). Sometimes they act unintentionally, or inadvertently (cf Zeelim A, 1990). The System-2 model assumes that the human element is rational, and entirely dedicated to operating by the book. This model is suited to describe the normal operation of low-risk systems. These assumptions do not apply to operating under stress, such as in multi-tasking (cf AF 447, 2009; WWII B-17). The human-factors version of

Murphy's Law is that the human element is error-prone: if the operators might fail, eventually they will.

Responsibility biasing

Traditionally, people regard the operators as responsible for preventing failure, and when they fail, the failure was attributed to the operator's errors. Donald Normal protested against this approach:

> Over 90% of industrial accidents are blamed on human error. You know, if it was 5%, we might believe it. But when it is virtually always, shouldn't we realize that it is something else?

https://jnd.org/stop_blaming_people_blame_inept_design/

The paradox of human errors

A primary reason for the popularity of the errors concept is vendor biasing. According to Erik Hollnagel, (Position Paper for NATO Conference on Human Error, 1983 https://erikhollnagel.com/onewebmedia/URTEXT%20on%20HE.pdf) errors are instances of normal operation with costly results.

This definition implies that by definition, the operators cannot prevent the errors, because if they prevented an error, then the error did not exist. Yet, the operators are typically regarded as accountable. The conclusion is that the term error is a bias, intended to divert the focus from the developers' mistakes to the operators.

The HSI challenge

According to many case studies, errors involve failure to notice exceptional situations, such as those due to a change in the operational mode (cf several TO/GA accidents).

A proactive version of Murphy's Law is about responsibility: instances of errors should be attributed to design mistakes: failure of situation awareness should be attributed to mistakes in the design of the human-machine coordination, not to the human operators. It is the developer's responsibility to prevent operator errors.

Usability vs. Safety

Human Factors Engineering is about considering human factors in the system development (design, verification, validation). The focus is on usability. The goal is a seamless operation. Sometimes, however, the seamless operation involves safety issues. Examples are usability problems due to unintentional activation of computer shortcut keys, or unnoticed assignment of default values on reset.

From human factors to HSI factors

HSI factors are complementary to human factors. HFE sets the usability goals, and HSI engineering is about ways to achieve these goals. Because the barriers to usability involve problems in the coordination between the human and the technical elements, the ways to implement usability is by HSI engineering.

The risks of implicit rules

Many accidents are attributed to unexpected operator behavior. This is often the case when the design relies on implicit rules, which the developers believed that the operators should follow.

Rule-based design

According to Leveson's STAMP, the system control should be constrained by rules defining proper operation, namely, enforcing operation in the performance envelope. This is possible only if the rules are defined explicitly, and the system can verify at run-time compliance with the rules. Rule-based design implies that the rules are defined explicitly, and implemented as safeguards in the system design.

Scenario-based modeling

Analysis of many accidents indicates a problem of inconsistent assumptions about the scenario, which was not defined explicitly, and therefore was not implemented in the interaction protocols. In many case studies, the operator assumed a scenario that did not match the operational mode. For example, many annoying problems in using home devices are due to enabling changing the device settings while in normal operation. Also, many accidents, such as Aero Peru 603, are due to applying maintenance-only procedures while in normal operation.

Scenario-mode pairing

The records of many accidents do not include data about the ways the operators or the system may trace the scenario, or how the system matched the scenarios with the operational modes. A common practice to work around this problem is to imply the scenario from the mode. It turns out that this practice is the root cause of many accidents, in which the implied scenario did not match the real scenario, which was projected from the contextual tasks.

From HSI to system integration

In an article submitter to the INCOSE HSI 2021 conference, I proposed a universal model of a digital twin, which may be used to control the system behavior. The proposed digital twins may operate according to the cybernetics principle of self-control, and Leveson's STAMP principle of rule-based design. In the tutorial proposal submitter to the INCOSE IS2022, I proposed to extend the model, to apply it to the integration of any STS.

The Controller-Service Integration (CSI) Model

The complexity of system integration may be resolved by examination of the interactions between system units. It seems that any integration may be described as a collection of interactions between two units, in which one of them is functional, providing services, and the second is a user of these functions, and also controls the use of these functions. This model complies with the STAMP paradigm proposed by Leveson. When employing this model, the controller is an abstraction of the human operators, and the service is an abstraction of a system. In other words, HSI is a special case of CSI.

Essentials of controller-service coordination

To cope with the complexity of exception handling, the services should cooperate with the controller:

- The services should provide the controllers with data essential for proper decision-making: a preview of the upcoming situation, and the potential effects of applying various options.
- The services should warn the controllers about critical activity, such as changing from normal to an exceptional situation, and about exception escalation

- The services should provide the controller with information that may facilitate the troubleshooting
- The services should rebound from erroneous control selection, and inform the controller about such instances.

Behavioral twins

A behavioral twin is a digital twin of the service behavior, used to detect exceptional situations, and to provide the data required for the service control.

A model of a behavioral twin may include six layers of design entities:

1. The basic layer is that of the system units as above
2. Each unit may have performance variables, used also as risk indicators
3. The situation of each unit, represented by state machines, describes attributes of functionality, availability, operation, hazards, etc. Situations are classified as normal or exceptional.
4. The controller-service activity is defined in terms of situation changes, and the risks associated with these changes. A change from a normal to an exceptional situation is classified as a hazard. Other risks indicators are about process variables, such as the time of processing or inter-machine state transition
5. The controller-service behavior is defined in terms of the service response to exceptional activity, such as automated shutdown, transition to safe-mode operation, or alerting.
6. Secondary risks, due to failure to detect or recover from a hazard

Cost-effectiveness

Twins development may be affordable if it is based on pre-defined profiles of operational rules, such as setting, maintenance, and safety backup.

Related articles

2008 - Standards for Defending Systems against Interaction Faults, DOI: 10.1002/j.2334-5837.2008.tb00908.x
Incose International Symposium, Utrecht, The Netherlands.

https://www.researchgate.net/publication/286091336_Standards_for_Defending_Systems_against_Interaction_Faults

This is a premature article, including an analysis of human-machine integration failure, and concluding about the need for standards.

2008 - Extended System Engineering - ESE: Integrating Usability Engineering in System Engineering, (A. Zonnenshain, A. Harel), DOI: 10.1002/j.2334-5837.2008.tb00899.x
The 17th International Conference of the Israel Society for Quality, Jerusalem, Israel

https://www.researchgate.net/publication/253025923_Extended_System_Engineering_-_ESE_Integrating_Usability_Engineering_in_System_Engineering

This is a first attempt to integrate the human operators into the system design, by considering the mutual effects of the system complexity and the operator's errors

---

2009 - Task-oriented System Engineering (A. Zonnenshain, A. Harel), DOI: 10.1002/j.2334-5837.2009.tb00982.x
INCOSE Annual International Symposium, Singapore.

https://www.researchgate.net/publication/286237486_Task-oriented_System_Engineering

This is the first attempt to propose that the system design should be dominated by the operator's intentions.

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

This is a preliminary work about the challenge of expecting and coping with the unexpected.

2011 - Comments on IEC 60601-1-8. Letter submitted to IEC/TC 62 working group.

This is about the astonishing finding that standards about human factors are protecting the developers from potential charges, rather than protecting society from possible errors.

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

This is the first case study about employing the first version of the guide to preventing errors, during car driving

---

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

This is a report about a study of 67 case studies, analyzed between 2010-2015, about ways to prevent errors.

2018 - Agile Error Proofing: a framework for adaptive resilience assurance. Abstract submitted to the EuroControl Flight Safety Forum, 2018

The forum rejected this proposal, believing that it was too technical.

2019 - Engineering the HSI. The first INCOSE HSI conference, (HSI2019) Biarritz, France.

The article, selected for a keynote presentation, is a first attempt to convert design ideas into engineering practices.

2020 - System Thinking Begins with Human Factors: Challenges for the 4th Industrial Revolution. in R.S. Kenett, R.S. Swarz and A. Zonnenshain (Eds), Systems Engineering in the Fourth Industrial Revolution: Big Data, Novel Technologies, and Modern Systems Engineering, Wiley. DOI: 10.1002/9781119513957.ch15

This is my view of how human factors should adapt to the 4th industrial revolution.

2021 - Towards Model-based HSI Engineering: A Universal HSI Model for Utility Optimization. The second INCOSE HSI conference, (HSI2021) San Diego, November (Virtual), Preprint.

This is the first attempt to integrate the various guidelines for error prevention into the concept of model-based digital twins.

2021 - Scenario-based modeling. DOI: 10.13140/RG.2.2.12834.35523

This article summarizes the conclusion from the analysis of several accidents due to a mismatch between the scenario, as perceived by the operator, and the system operational mode, as defined in the requirements documents.

More: https://avi.har-el.com/eng/Articles/Articles.html

About: https://avi.har-el.com