Integration-Centered Design

A framework for preventing human errors

Avi Harel, Ergolight, +972 54 453 4501, ergolight@gmail.com

This paper targets developers of utility-critical systems, namely, safety or mission-critical systems, productivity-critical systems, home, and personal devices, as well as marketing and sales websites.

A primary hurdle to maximizing the system utility is the difficulties that the operators experience when approaching the performance boundaries. The reason for this is that regular training targets normal conditions. During normal operation, the operators encounter exceptional situations only occasionally, which is not sufficient for effective learning. Whenever they encounter an exceptional situation, they waste too much time trying to find their way around it. For example, informal studies on productivity in text editing indicate that about half of the time is wasted in recovery from errors.

Human errors are often due to difficulties of the human operators to handle problems in the integration of technical components. To prevent the errors, we employ a model of rule-based client-server integration. Eventually, human-system integration is a special case of client-server integration, and therefore, this model may also solve critical usability problems.

It might be too expensive to develop models dedicated to specific systems. Fortunately, in a prior study on system resilience, we have defined several patterns of handling exceptions. The conclusion is that we can formulate a universal hyper-model, consisting of layers of generic mini-models (GMM), in terms of operational rules. Model-based design enables seamless adaptation to design changes. Rule-based models enforce mitigating the risk of operational complexity. A universal hyper-model proposed here consists of seven layers, such that each layer consists of several GMMs:

- Structural layer: recursive definition, in terms of users, operators, and subsystem
- Functional layer: The operator's tasks and the potential risks
- Performance layer: potential failures due to diverting from the performance envelope
- Static layer: representation of the operational situations
- Dynamic layer: representation of the operational activities
- Behavioral layer: representation of the responses to events
- Resilience layer: representation of safety backups

Each of the GMMs depends on GMMs from the previous layer. For each of the GMMs, we describe the main entities, with a focus on the operational rules. A key concept is of scenarios, which enable reducing the complexity of situational transitions by state encapsulation. Scenario-based design should enable direct mapping from intentions to actions.

Utility-critical systems should incorporate means, including sensors and data analytics, for informing the operators and the developers about integration flaws. A universal architecture of client-server integration may enable the implementation of this model.

This architecture may employ trend analysis by digital twins, used to support critical usability challenges such as providing previews of upcoming situations, evaluation of decision options, and assisting in troubleshooting.