Scenario-driven operation: the BrahMos case study

Avi Harel, Ergolight

ergolight@gmail.com

Situational consistency

During the operation, various states in various components need to change to fit the functional needs. A situation that complies with the functional needs is regarded as consistent with respect to the supported function.

Operational reliability relies on situational consistency. The operation might fail when the situation is not consistent with a function.

The BrahMos case study

BrahMos is a supersonic land attack missile with nuclear capabilities. On 9 March 2022, during routine maintenance, a land-based version of BrahMos was fired accidentally into Pakistan. From its initial course, the object suddenly maneuvered towards Pakistani territory and violated Pakistan's air space, ultimately falling near Mian Channu.

Investigation

On 12 March 2022, the Foreign Office of Pakistan issued a statement demanding "a joint probe to accurately establish the facts surrounding the incident" while rejecting New Delhi's decision to hold an internal inquiry. India also said that they have ordered a high-level Court of Enquiry to look into the incident.

On 23 August 2022 the IAF initiated a Court of Inquiry [CoI] to probe the misfire, attributing the missile's firing to 'several omissions and commissions' by three officers on its Combat Team. The officers were primarily held responsible for the incident for "deviation from the Standard Operating Procedures" and their services were terminated by the Indian government with immediate effect. These officers are currently challenging the findings of the CoI in the Delhi High Court.

Later, the Pakistani Ministry of Foreign Affairs issued a statement rejecting the Indian investigations. Stating that "systemic loopholes and technical lapses of serious nature in handling of strategic weapons cannot be covered up beneath the veneer of individual human

error", Pakistan maintained its demand for a joint investigation of the incident in the spirit of transparency.[1]

For two years the IAF did not disclose the investigation results. The appeal to the Delhi High Court creates an opportunity to learn about the circumstances of the incident.

Root Cause Analysis (RCA)

The system consistency depends not only on the function, but also on the operational scenario. If different components assume different scenarios, then the situation might not be consistent (cf. Harel, 2021).

The missile comprises two essential parts: combat connectors and the junction box.

Combat connectors are interfaces that facilitate communication between the missile system and the control mechanisms of its launcher. They make possible command inputs, status monitoring, and signal activations possible. On the other hand, the junction box is a critical connectivity hub for data and electrical links. This component is vital for adjusting the missile's flight path and targeting based on commands or fresh intelligence information.

Maintenance or upgrade transportation scenarios often require temporary alternate connections for diagnostic examinations, system tests, or software updates. Strict safety measures are continuously implemented to prevent accidental activation of the missile's systems.

Typically, fail-safe procedures involve multiple authorization codes, electronic locks, and physical deactivation when not in active use, significantly reducing the potential for inadvertent discharge.

The engagement or activation of these combat connectors occurs at specific moments throughout the missile's preparation and flight phases. They play a vital role in ensuring the initial targeting data, system checks, and status updates reach the missile. Even post-launch, these connectors continue to provide real-time updates and adjustments based on mid-flight modifications.

Despite the combat crew being aware that the combat connectors of the missiles were connected to the junction box, they did not prevent the Mobile Autonomous Launcher commander from launching the Combat Missile.

Operational errors

Analysis of many accidents has shown that the term human error is just a name for operational failure that the human operator was not able to prevent (cf. Dekker, 2007). To eliminate human errors, we need to understand how the operation fails. The challenge is to get enough evidence to understand how errors are generated. In the BrahMos' example, the system design enabled three errors and two design mistakes, which together enabled the unexpected launch. The errors are:

- State confusion: The Combat Team selected the *'live state'* instead of the *'inert state'*, which was adequate for the 'inspection' scenario, due to misunderstanding of the semantics of '*live state*'.
- Connection confusion: The combat connectors remaining attached to the junction box to maintain flow of data required for subsequent operation.
- The combat team, fully cognizant that the missile's combat connectors were linked to the junction box, did not take action to prevent the Mobile Autonomous Launcher Commander from initiating the firing of the missile.

The design mistakes are:

- Safety relied on a complicated security system with several layers of authentication, enabling bypassing them
- The design relied on situation perception and the reasoning of the team members and did not prevent the launch in this exceptional situation.

RCA conclusions

The system consistency depends not only on the function, but also on the operational scenario. If different interacting components assume different scenarios, then the situation might not be consistent.

The accident of this case study is due to failure to maintain situation compliance following a change of the operational scenario from operational to testing. This kind of accident is typical of systems that do not maintain the primary scenario variable. In such cases, the scenarios are fuzzy, and consequently different system components might assume different scenarios. In such cases, the situation is not consistent.

Enforcing situational consistency

To enforce situational consistency, the system should adapt to scenario changes. We may distinguish between two types of situational variables: controllable variables, such as state variables, and uncontrollable variables, such as continuous variables, obtained by measurements.

In an earlier study about the unintentional launch of the Cheongung missile it was found that the root cause was inconsistent system configuration. The solution proposed there was to define and apply rules for associating the configuration with the operational scenario (Harel, 2024).

In a scenario-driven operation design we deal with all kinds of scenario dependent factors. In this case study we deal with two factors of scenario compliance: one with the configuration compliance, and the other with the operational mode.

Situational rules

To enable the situation adaptation, the scenario should be defined explicitly, and implemented as a primary system variable. Otherwise, different system components might assume different scenarios, and the corresponding situations might be different. The scenarios should be defined such that a scenario change involves:

- Enforcing consistent change of the adaptable variables
- Verification of the consistency of unadaptable variables.

In this case study, the enforcement is applicable to the mode change, such that in the ‘inspection’ scenario should enforce the ‘inert state’. On the other hand, in this case, we cannot enforce physical cable connection by software, but we can enforce verification, and alerting on violation of the situational rules. The situational rules applicable to this case study are:

Scenario-driven setting rules

- The system should notify and disable launching the combat missile when in ‘live state’ when the scenario is ‘inspection’

Configuration rules

- Whenever the combat connectors are connected to the connection box and the scenario is ‘inspection’, the system should notify the exception and disable the Mobile Autonomous Launcher commander from launching the combat missile.

Enforcing the situational rules

To enable the situation verification, the system requirement should specify the operational scenarios, the system design should manage the scenarios, and the software program should verify that the implementation complies with the design. The focus of situation verification is on the mode compliance with the scenario according to the rules and on disabling the operation when they do not comply with the scenario. The design challenge is to specify the rules defining the scenario-mode compliance and the reaction to violating these rules.

Activity rules

Activity may be defined in terms of changes in the system situation. Accordingly, activity rules are about situational changes. The activity rule applicable to this case study is:

On change to the ‘inspection’ scenario, the system should:

- Apply the scenario-driven rule: enforce the ‘inert state’, and
- Prompt the operators to disconnect the combat connectors from the connection box.

Error proofing

To be on the safe side, we should protect the system from all risky situations, because we cannot tell when one of them might be disastrous. Practically, this implies that error proofing ought to be a key topic of systems engineering.

Enforcing situation awareness

To enforce the operator’s awareness of the rule violation we need to apply two mechanisms:

- Situational verification: ongoing notification, as long as the configuration does not comply with the rules
- Activity verification: an alarm on changing from normal to exceptional configuration.

Conclusion

The BrahMos accident demonstrates a need for early detection of configuration errors and mode errors. The method demonstrated here is based on rules for associating the configurations and modes with the scenarios.

The vision proposed here is that system engineering standards may include a chapter on when and how to apply the method for scenario-driven operation.

References

Dekker, S 2007. Just Culture: Balancing Justice with Accountability.

Harel, A 2021. Scenario-based modelling. DOI: 10.13140/RG.2.2.12834.35523

Harel, A 2024. Configuration verification: the Cheongung case study.
DOI: 10.13140/RG.2.2.27364.18564