

From Trust Repair to Interaction Repairability: A Conceptual Framework for Recoverable Generative AI Experiences

Zulisman Maksom^{1*}, Che Ku Nuraini Che Ku Mohd², Faaizah Shahbodin³

^{1,2}Faculty of Information and Communication Technology, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

³Pervasive Computing & Educational Technology, Department of Media InteractiveFakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka

DOI: <https://doi.org/10.47772/IJRISS.2026.100800927>

Received: 2 September 2026; Accepted: 7 September 2026; Published: 21 September 2026

ABSTRACT

Generative AI is now part of everyday writing, coding, design, education, and decision-support work, yet the interfaces around these systems are still weak when an output goes wrong. Research on reliability, trust repair, explainability, controllability, mental models, and conversational repair explains important parts of the problem, but gives less attention to a practical question: how should an interface help a user recover the work that has been affected? This conceptual paper introduces the Repairable Generative Interaction (RGI) Framework. It defines interaction repairability as the degree to which a human-AI system helps users detect a consequential breakdown, identify the affected objects and dependencies, understand the mismatch between intended and generated states, make a bounded correction, restore a coherent task state, and retain a traceable record of the recovery. The framework was developed through a problem-driven integrative synthesis of contemporary human-AI research and established HCI work on error recovery, undo, reversibility, and user control. RGI brings together six proposed failure classes, six recursive repair functions, four interaction layers, and five cross-cutting design commitments. It also distinguishes repairability from reliability, explainability, controllability, resilience, contestability, and trust repair, and proposes measurable indicators and seven testable propositions. A worked example from Malaysian open and distance learning shows how the framework can be applied without presenting it as empirical evidence. RGI remains a conceptual framework and now requires empirical testing across task recovery, user agency, mental-model accuracy, trust calibration, accessibility, cultural-linguistic variation, and governance contexts.

Keywords: generative AI, human-AI interaction, interaction repairability, error recovery, user control, explainable AI

INTRODUCTION

Generative AI has moved quickly into everyday work. People now use it to produce text, images, code, designs, summaries, and recommendations. The conversational interface makes this work look straightforward: a user states what is needed, receives an output, and refines it. The difficulty becomes clearer when something goes wrong. A model may misunderstand the task, introduce an unsupported claim, alter approved content, ignore a constraint, or carry out an action the user did not intend. In many current systems, the response is little more than an apology, regeneration, or another round of prompting. The user is left to work out what failed, what changed, and how to recover the original task.

These failures are not only questions of model accuracy. Reliability concerns whether a system performs within expected limits; recovery concerns what users can do after those limits have been crossed. HCI has dealt with this problem for decades through work on error handling, reversibility, undo, and user control. [21] argued that design should reduce both the likelihood and the consequences of human error. Early studies of undo likewise treated reversal as part of interaction recovery rather than as a purely technical command [33], [2]. Generative AI makes recovery harder because one instruction can alter a large, semantically connected

artifact, while the relationship between prompt, inference, evidence, and output is often difficult for users to inspect.

Related human-AI research offers several useful pieces of this picture. Trust research examines appropriate reliance and how trust changes after an error [20], [19], [32]. Explainable AI asks whether explanations improve understanding, mental models, and decision quality [14], [17], [26]. Studies of controllability and decision control show that user influence matters, especially when it is supported by explanation and task context [29], [31]. Conversational repair research examines how breakdowns are noticed and resolved jointly [3], [12]. What remains less developed is an account of the interface support needed to restore work after a generative breakdown.

To address that gap, this paper treats interaction repairability as a distinct system-level UX quality. It refers to the extent to which a human-AI system allows users to detect a consequential breakdown, identify the affected objects and dependencies, understand the difference between intended and generated states, make a bounded correction, restore a coherent task state, and keep a traceable account of what happened. The purpose is not to make the system seem trustworthy after a failure. The purpose is to give the user a clear and controllable route from an uncertain or damaged state back to work that is safe, coherent, and accountable.

The paper makes four contributions. It first connects interaction repairability to both foundational HCI research on recovery and recent human-AI work. It then defines the internal structure of RGI through six failure classes, six recursive repair functions, four interaction layers, and five cross-cutting design commitments. A third contribution is a clearer conceptual boundary between repairability and related ideas such as reliability, explainability, controllability, resilience, contestability, and trust repair. Finally, the paper proposes an operationalisation model and seven propositions that can be tested against outcomes such as task restoration, user agency, mental-model accuracy, trust calibration, and accountable system use. No user study, dataset, or empirical result is reported here.

Conceptual Background

This section draws together the bodies of literature that become most relevant once a generative interaction has already broken down. The focus is practical as well as theoretical: what does a user need from the interface when the correctness, integrity, or accountability of ongoing work is in doubt?

Classical HCI foundations of recoverability

Recoverability has a long history in HCI. Early work on human error argued that good design should reduce not only how often errors occur but also the damage they cause [21]. Later work examined forward recovery, backward recovery, and reversibility in interactive systems. Undo has been described as an interactive recovery facility [33], while other work argues that undo should be understood from the user's intention rather than only from the system command [2]. That distinction is especially relevant to generative AI. A technical undo may reverse an operation, but it may still leave the user unsure about what changed, why it changed, or which parts of the work were affected.

The same principle appears in work on selective undo. Research has shown that users can benefit from reversing one action while keeping later work where this remains meaningful [8]. Generative editing creates a similar need: a user may want to correct one claim, paragraph, function, visual region, or constraint without regenerating the entire artifact. Other HCI repair research has also identified undo, redo, and new requests as recurring elements of repair in human-computer interaction [11]. These studies make an important point for RGI. Recovery itself is not new; the newer challenge lies in the scale, opacity, semantic coupling, and probabilistic behaviour of generative systems.

RGI builds on this earlier recovery tradition rather than claiming to replace it. Its focus is on generative interactions in which the object being repaired may be a claim, artifact, task state, evidence chain, instruction set, or accountability relation. In such settings, reversal alone is often not enough. Users may also need

localisation, interpretation, provenance, protection of approved constraints, and a route to human escalation when the system cannot resolve the problem safely.

From reliability to recoverability

Reliability and repairability address different moments in system use. Reliability is concerned mainly with whether behaviour remains correct or acceptable. Repairability becomes important once a deviation has occurred or is suspected. Human-centered AI research argues for combining automation with strong human control [28]. RGI adds a post-failure question to that position: once control is needed, what path does the interface give the user for recovering the task?

Evidence from generative-AI use shows why this matters. Users have reported misinformation and response-quality problems in widely used systems [1], while evaluations of generative-AI interfaces continue to identify weaknesses in user control and error handling [7]. A regenerate command may give a different answer, but it does not necessarily identify the fault in the previous answer, preserve unaffected work, or make clear which version should be treated as authoritative. Recovery therefore depends on the interface as well as the model.

Recent empirical work shows that these difficulties are visible in user behaviour, not only in conceptual debates. A study of 312 semi-technical users found that perceived explainability and intention alignment were strongly associated with perceived reliability and trust, while the relationship involving perceived agency was more modest [13]. Interviews with 29 ChatGPT users likewise showed that trust judgments draw on several sources at once, including information quality, system quality, technology ethics, organisational reputation, and cultural context [34]. Semester-long human-GAI team collaboration showed trust, distrust, and ambivalence rather than a simple high-or-low trust response [22]. Post-error studies make the recovery problem even clearer. Guilty apologies were found to improve trust repair and perceived experience without improving perceived agency [32], while explanation-based and apology-based strategies were shown to repair subjective trust but differed in recovery effort and actual continuance behaviour [19]. In an adjacent real-world chatbot setting, a randomized field experiment found that collaborative repair resolved more conversational breakdowns and mitigated negative customer outcomes [12]. Taken together, these findings suggest that reassurance, understanding, agency, behavioural continuation, and restoration of the affected work are related but distinct recovery outcomes.

Trust repair and appropriate reliance

Trust in AI is relational and task-dependent rather than a simple global attitude [20]. Recent work has examined trust calibration in everyday generative-AI use [4], cultural dynamics in AI trust [10], and how anthropomorphic or emotional cues influence trust formation [13], [15], [30]. After a failure, apology and explanation can affect subjective trust or continuance decisions [19], [32]. However, restored confidence does not guarantee restored work.

Within RGI, trust calibration is treated as a possible result of recovery rather than as the object of repair itself. A successful repair may increase trust, reduce it, or leave it unchanged. Any of these outcomes can be appropriate if the user's reliance becomes better aligned with what the system can actually do and the affected task is left in a safe and understandable state.

Explanation, mental models and actionability

Explainability is most useful when it helps people understand a situation and decide what to do next, rather than when it adds transparency for its own sake [14], [26]. Its value also depends on timing, context, and the task at hand [23], [24]. During repair, users often need a local explanation: what failed, where it happened, what the system inferred, what evidence supports that inference, and what can be changed safely.

Mental models matter for the same reason. After a breakdown, users may need to revise what they believe the AI can and cannot do. Recent scoping work shows that research on human-AI mental models remains conceptually diverse and sensitive to context [9]. RGI treats this learning as part of closing the repair episode.

The user should be able to see what changed, what is still unresolved, and which constraints or lessons will carry forward.

Control, collaborative repair and the remaining gap

Controllability gives users some influence over what a system does next, but that influence is only useful when they can see what can be changed and understand the likely consequences. Research shows that control can interact with explanation to shape perceptions and compliance [29], [31]. Even so, a prompt box does not by itself provide a reliable way to restore work that has already been damaged.

Recent collaborative-repair research offers the closest contemporary parallel in customer-chatbot interaction [12]. That work shows the value of involving both parties in resolving a breakdown, but its main object is conversational misunderstanding. RGI broadens the unit of repair to include generated artifacts, task states, evidence, and accountability. A distributed-cognition account similarly explains how human-AI coordination can break down [35], whereas RGI concentrates on the interface support needed after that breakdown has been recognised.

Conceptual boundary of interaction repairability

RGI's contribution lies in the object and mechanism of recovery rather than in a claim that related concepts are inadequate. Reliability, explainability, controllability, resilience, contestability, and trust repair remain important, but each addresses a different part of the problem. RGI brings these concerns together around a narrower question: can the user recover the affected work in a way that is local, understandable, reversible where possible, traceable, and appropriately governed?

Table 1: Conceptual boundary between interaction repairability and adjacent constructs

Construct	Primary question	Temporal focus	Primary object	Typical outcome	Boundary from RGI
Reliability	Will the system behave correctly?	Before/during failure	System performance	Fewer or less severe failures	RGI addresses what the user can do after a breakdown.
Explainability	Why did the system produce this output?	Before/after output	Model, decision or output	Understanding	RGI uses explanation selectively when it enables diagnosis and correction.
Controllability	Can the user influence system behaviour?	During interaction	System behaviour	Influence over next action	RGI adds protection and restoration of prior task state.
Resilience	Can the technical system continue or recover?	After disruption	Technical system	Operational continuity	RGI focuses on human-understandable restoration and accountability.

Contestability	Can a decision be challenged and reviewed?	After consequential decision	Decision/governance process	Review or appeal	RGI includes local repair as well as formal escalation.
Trust repair	Can confidence or willingness to rely be restored?	After trust violation	User-system relationship	Changed trust/reliance	RGI targets restoration of affected work; trust may rise or fall appropriately.
Interaction repairability	Can users diagnose, correct and restore affected work?	After or during breakdown	Interaction/task/artifact state	Safe, traceable task recovery	Central construct proposed by RGI.

Adjacent frameworks and novelty boundary

Table 2 places RGI alongside the recent frameworks and studies that are closest to its scope. The comparison is deliberately conservative. RGI is presented as an integration of repair objects, recursive recovery functions, task-state restoration, provenance, constraint preservation, and escalation, not as the invention of repair itself.

Table 2: RGI versus adjacent repair and human-AI frameworks

Adjacent work	Primary object	Core scope/mechanism	Boundary from RGI
Gnewuch & Reinkemeier [12]	Conversational breakdown	Collaborative repair of intent-recognition failures in chatbots	RGI extends repair to artifacts, state, provenance, constraints and rollback.
Konopka & Wiesche [19]	Post-error trust	Human-like versus system-like trust-repair strategies	RGI evaluates whether work can be recovered, not only whether trust changes.
Xu et al. [32]	Trust and relationship repair	Apology and mind perception	Apology may support relational repair but does not replace diagnosis or state restoration.
Araujo [4]	Trust calibration	Dynamic trust calibration in everyday GenAI use	RGI treats calibrated trust as an outcome or consequence of recovery.
Zhao & Han [35]	Human-AI cognitive coordination	Distributed-cognition account of coordination breakdown	RGI specifies interface support after coordination has failed.
Batool & Hussain [7]	GenAI GUI usability	Usability and ethical evaluation of GenAI interfaces	Provides evidence of interface problems; RGI

			proposes a recovery architecture.
Classical HCI undo/recovery	Interaction error	Reversal, selective undo, recovery intention	RGI extends recovery to probabilistic, semantic and governance-sensitive generative work.

Framework Development Method

Conceptual synthesis approach

RGI was developed through a problem-driven integrative conceptual synthesis. The purpose was to build theory rather than estimate effects, map prevalence, or claim an exhaustive systematic review. The synthesis was used to identify recurring breakdowns in generative interaction, connect them with established HCI and human-AI concepts, and work out the minimum interface capabilities needed for recovery.

The original opportunity-mapping stage retained 30 journal articles and one relevant preprint. A later round of targeted retrieval added foundational HCI work needed to strengthen the theoretical basis of the framework. The final synthesis corpus comprised 35 peer-reviewed scholarly works and one preprint. Scopus and Web of Science (WoS) were the only primary databases used for literature identification. Crossref and official publisher websites were used separately to verify bibliographic details and DOI information.

Search logic and source selection

The search was carried out in Scopus and Web of Science (WoS). These databases were chosen because they provide broad coverage of peer-reviewed work in HCI, information systems, artificial intelligence, and human-AI interaction. Search combinations paired terms such as "generative AI" and "human-AI interaction" with "error recovery", "interaction repair", "trust repair", "explainability", "controllability", "mental models", "undo", and "reversibility".

Citation chaining was also used where recent work pointed back to established HCI sources relevant to repair and recoverability. Crossref and publisher websites were used only for metadata and DOI verification, not as additional literature databases.

Table 3: Inclusion, exclusion and synthesis criteria

Primary databases	Scopus and Web of Science (WoS) only. Crossref and official publisher pages were used for metadata and DOI verification, not as literature-search databases.
Criterion	Operational rule
Inclusion	Peer-reviewed HCI, human-AI interaction, information systems, UX, or AI studies addressing breakdown, recovery, trust repair, explanation, control, mental models, provenance, or task-state recovery; foundational HCI studies directly relevant to error recovery and undo.
Exclusion	Purely algorithmic error-correction studies without a human-interaction implication; general AI-ethics commentary without a recovery mechanism; duplicate records; papers whose relevance could not be established from verified metadata or accessible abstract/full text.

Temporal logic	Recent GenAI evidence was prioritised for the changing technological context, while older HCI sources were deliberately retained where they established durable principles of recovery and reversibility.
Synthesis unit	Conceptual mechanism rather than effect size: object of failure, recovery action, user-control provision, evidence/explanation provision, state-restoration mechanism, and governance/escalation mechanism.
Novelty test	A candidate concept was retained only when it added a distinct recovery object or mechanism beyond adjacent trust, explanation, coordination, and conversational-repair frameworks.

Coding and framework construction

Each retained source was read with six questions in mind: What is the object of failure? How is the failure detected? What information helps a user diagnose it? What forms of correction or control are available? How is a prior or safe state restored? What evidence, memory, or accountability information remains after recovery? This coding kept the synthesis focused on recovery mechanisms rather than on trust, usability, or model performance in general.

The framework emerged through several rounds of comparison and abstraction. Similar breakdowns were grouped into provisional failure families, while recovery actions were separated from the design properties that make those actions possible. The analysis also distinguished where repair takes place, which led to the four interaction layers. These elements were repeatedly checked against the source material and against nearby concepts to reduce unnecessary overlap.

The novelty claim is deliberately narrow. The search did not identify a substantially equivalent cross-interface framework that combines generative failure objects, recursive repair functions, task-state restoration, provenance, constraint persistence, and governance within one UX model. This does not establish absolute novelty, and the paper does not make that claim. It identifies a specific integration that appears to be missing from the literature reviewed.

The Repairable Generative Interaction Framework

Definition and unit of analysis

RGI treats interaction repairability as a system-level UX quality that emerges from the relationship among the interface, model behaviour, artifact state, user intent, evidence, and institutional accountability. The unit of analysis is the repair episode: a period that begins when a consequential breakdown is noticed or suspected and ends when the user reaches a safe, coherent, and traceable task state, or when the issue is escalated because the system cannot resolve it safely.

Failure classes

The framework proposes six failure classes. They are intended as useful analytical families rather than as an empirically complete or mutually exclusive taxonomy. Semantic misalignment occurs when the system interprets the task differently from the user. Epistemic failure concerns unsupported, incorrect, or untraceable claims. Normative mismatch appears when an output conflicts with policy, values, or cultural expectations. Procedural failure involves an incorrect action or sequence. State or artifact failure occurs when a local request produces unintended wider changes or damages a useful version. Relational failure concerns social cues, responsibility, or anthropomorphic language that makes accountability less clear.

More than one class can appear in the same episode. A fabricated policy statement, for example, may be both epistemic and relational, and an attempt to correct it may create a state failure if the system rewrites content that was already approved. The categories are useful only when they lead to different recovery needs. Epistemic problems call for evidence and provenance; state failures call for rollback or version restoration.

Six recursive repair functions

The six RGI stages are treated here as repair functions, not as a fixed sequence. A repair episode may move forward, skip a function that is not needed, or return to an earlier one. A correction may expose a new problem and send the user back to Signal or Locate; a failed interpretation may require more evidence before correction can continue. This recursive view better reflects how recovery is likely to happen in real generative work.

Signal: A breakdown becomes visible through user observation, system detection, external evidence, or policy. A useful signal is granular enough to identify the affected object and distinguishes uncertainty from confirmed error.

Locate: The interface links the breakdown to a specific claim, component, action, artifact region, or decision and, where relevant, exposes dependencies so that the user can estimate the spread of impact.

Interpret: The user and system compare intended and generated states. The interface provides enough evidence to diagnose the discrepancy without forcing the user to inspect irrelevant technical detail.

Co-correct: Correction is bounded and negotiable. The user can target the affected object, modify a constraint, accept or reject a proposed fix, or refer the issue to a human authority. System suggestions and user decisions remain distinguishable.

Restore: The task returns to a coherent and traceable state. Restoration may involve rollback, selective reapplication, revalidation, or confirmation that approved content and constraints remain intact.

Learn: The repair closes with a concise account of what changed, what remains unresolved, what evidence was used, and what user-approved constraints should persist. Retained memory must remain visible, editable, and removable.

Cross-cutting design commitments

Five design commitments support these repair functions across different interface types. Bounded reversibility allows important actions to be undone within a stated scope. Local editability lets the user change one part without replacing the whole artifact. Provenance persistence keeps sources, actions, and dependencies traceable. Constraint persistence protects requirements that the user has already approved. Proportionate escalation provides a route to appropriate human authority when the risk or uncertainty is beyond what the interface should handle on its own.

Table 4: RGI design commitments and interface provisions

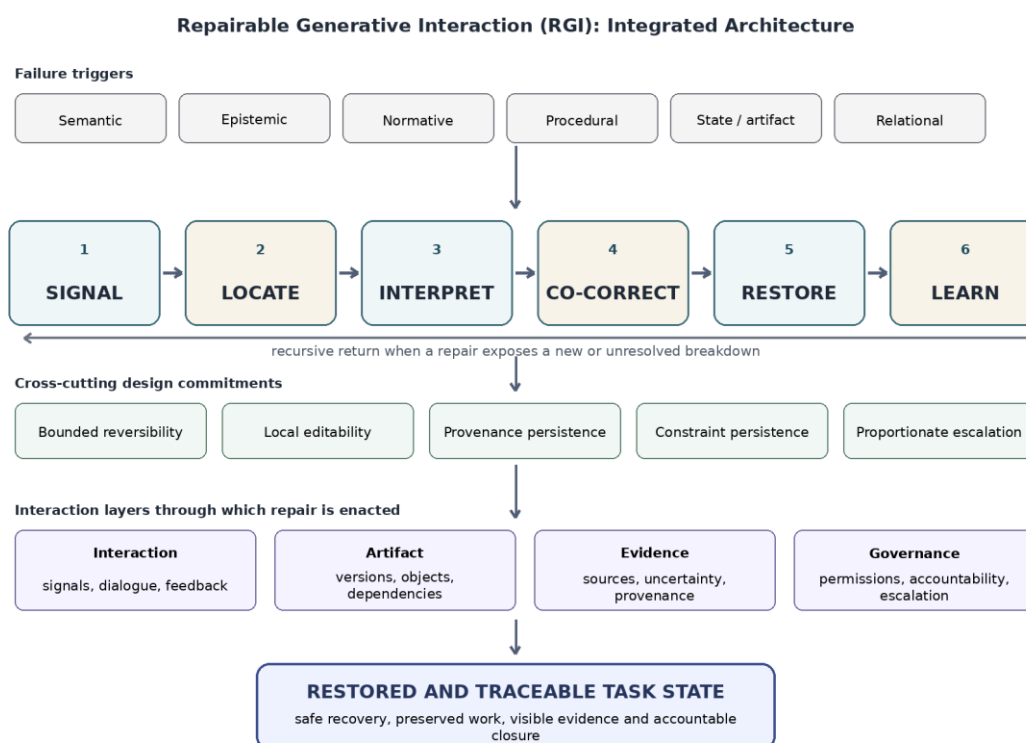
Commitment	Design question	Illustrative provision
Bounded reversibility	Can the affected action be undone without creating greater harm?	Local rollback, approved checkpoints, version comparison, pre-apply warning.
Local editability	Can the user modify only the affected object?	Claim-, paragraph-, region-, function-, or parameter-level correction.

Provenance persistence	Can the user reconstruct what changed and why?	Source links, action logs, model/action history, dependency traces.
Constraint persistence	Do approved requirements survive repair?	Visible constraint ledger, locked requirements, editable project memory.
Proportionate escalation	Can unresolved or high-stakes risk reach the right authority?	Role-based hand-off with evidence, state, decisions, and unresolved issues.

Interaction layers

Repair can involve four layers. The interaction layer covers signals, dialogue, explanations, and feedback. The artifact layer covers generated objects, versions, dependencies, and task state. The evidence layer covers provenance, source quality, uncertainty, and verification. The governance layer covers permissions, responsibility, escalation, and accountability. A repair may touch one layer or several at the same time.

Figure 1: Integrated architecture of the Repairable Generative Interaction Framework



Note: The six repair functions are recursive rather than mandatory sequential steps. Failure classes can co-occur, and the four interaction layers and five design commitments jointly condition whether a coherent task state can be restored.

Functional architecture

Table 5 summarises the strongest expected links between failure classes, repair functions, interaction layers, and design commitments. These are not one-to-one rules. They indicate which parts of the framework are likely to matter most for a given type of breakdown and can be tested or revised in later empirical work.

Table 5: Functional architecture of RGI

Failure class	Dominant repair functions	Primary layer(s)	Critical commitment(s)
Semantic misalignment	Signal, Interpret, Co-correct	Interaction	Local editability; constraint persistence
Epistemic failure	Signal, Locate, Interpret, Restore	Evidence	Provenance persistence
Normative mismatch	Interpret, Co-correct, Restore	Governance / Evidence	Proportionate escalation; provenance persistence
Procedural failure	Locate, Co-correct, Restore	Interaction / Artifact	Bounded reversibility; local editability
State/artifact failure	Locate, Restore, Learn	Artifact	Bounded reversibility; constraint persistence
Relational failure	Signal, Interpret, Learn	Interaction / Governance	Proportionate escalation; provenance persistence

Operationalising Interaction Repairability

For empirical work, interaction repairability is best treated initially as a formative, multidimensional UX construct rather than as a single attitude scale. The dimensions below describe different aspects of recovery performance. They should not yet be interpreted as a validated instrument; their structure, weighting, and discriminant validity remain questions for future studies.

Table 6: Proposed dimensions and indicators of interaction repairability

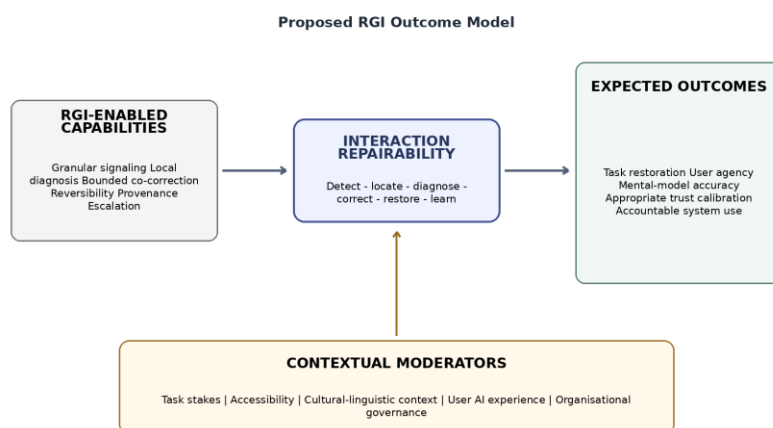
Dimension	Operational question	Illustrative indicators
Breakdown detectability	Can users recognise a consequential failure?	Detection accuracy; detection latency; missed-failure rate
Localisability	Can users identify the affected object and spread?	Correct object localisation; dependency identification accuracy
Diagnostic clarity	Can users explain the discrepancy between intended and generated states?	Diagnosis accuracy; confidence calibration; unresolved ambiguity
Correction precision	Can the fix be restricted to the intended scope?	Collateral-change rate; proportion of unaffected content preserved
Restoration success	Can the user return to a coherent safe state?	Restored-task proportion; successful rollback/revalidation; recovery attempts

Traceability	Can the user reconstruct what changed and why?	Provenance recall; source verification success; audit-trail completeness
Recovery efficiency	How much effort is required to recover?	Time to safe state; number of actions; subjective recovery workload
Escalation appropriateness	Does the system route unresolved risk correctly?	Correct escalation decision; completeness of hand-off context
Learning persistence	Does recovery reduce recurrence while preserving user control?	Repeated-error rate; visibility/editability of retained constraints

Proposed Outcome Model and Propositions

The outcome model separates three things that were less clearly distinguished in the original version: the interface capabilities provided by RGI, interaction repairability itself, and the outcomes that may follow from a successful or unsuccessful recovery. RGI-enabled features are expected to strengthen repairability, which may in turn affect task restoration, user agency, mental-model accuracy, trust calibration, and accountable system use. These relationships may differ according to task stakes, accessibility needs, cultural-linguistic context, user experience with AI, and organisational governance.

Figure 2: Proposed RGI outcome model



Note: The arrows represent proposed relationships for future testing, not demonstrated causal effects. Trust calibration is treated as an outcome of recovery quality rather than the primary repair target.

Proposition 1: Greater breakdown-signal granularity will increase failure-detection accuracy and reduce detection latency compared with generic system-level warnings or apology-only responses.

Proposition 2: Greater diagnostic specificity will increase fault-localisation accuracy and reduce unnecessary regeneration of unaffected task components.

Proposition 3: Local co-correction controls will reduce collateral artifact change and increase perceived user agency compared with full-output regeneration.

Proposition 4: The combined availability of bounded reversibility and persistent provenance will increase restoration success and reduce unresolved recovery errors relative to either mechanism alone.

Proposition 5: Repair explanations that explicitly connect intended state, generated state, evidence, and corrective options will improve post-repair mental-model accuracy more than apology-only responses.

Proposition 6: Stake-sensitive escalation will increase the appropriateness of human intervention and reduce unresolved high-severity failures compared with uniform escalation rules.

Proposition 7: Visible and editable persistence of user-approved constraints will reduce recurrence of previously corrected failures across subsequent interactions without reducing perceived user control.

Worked Illustrative Case: Malaysian Open and Distance Learning

Consider a lecturer in a Malaysian open and distance learning setting who uses generative AI to align learning outcomes, assessment rubrics, and instructional materials with university policy. The AI produces a plausible draft but inserts a policy rule that does not exist. When the lecturer asks for a correction, a conventional chat interface apologises and rewrites the unit. The false rule disappears, but two previously approved rubric criteria and an accessibility instruction are also changed. The lecturer now has to determine which version is trustworthy and which changes were unintended.

An RGI-enabled interface would handle the episode more selectively. It would flag the unsupported policy claim at claim level, identify the rubric cells and instructions that depend on it, and compare the generated interpretation with the relevant policy source. The lecturer could then correct only the affected statement, preserve unrelated approved content, or mark the issue unresolved and refer it for institutional confirmation. Once the repair was complete, the interface would record what changed and retain an editable constraint requiring future policy claims to be checked against an official source. This example is used only to show how the framework works. It is not an evaluated UTeM system, a participant study, or evidence of an observed outcome. Its value lies in making the repair object, evidence requirement, state-restoration need, and governance boundary visible in a familiar educational task.

Table 7: Mapping the ODL scenario to RGI

Scenario event	Repair object	Failure class	RGI function	Layer	Commitment
Initial breakdown	Unsupported university-policy claim	Epistemic	Signal + Locate	Evidence	Provenance persistence
Collateral rewrite	Approved rubric/accessibility content changed	State/artifact	Locate + Restore	Artifact	Bounded reversibility
Intent correction	Only disputed statement should change	Semantic/procedural	Interpret + Co-correct	Interaction	Local editability
Authority uncertainty	Policy interpretation remains unresolved	Normative/governance	Co-correct + Restore	Governance	Proportionate escalation
Future continuity	Verified-source rule should persist	Potential recurrence	Learn	Interaction/Evidence	Constraint persistence

DISCUSSION

Theoretical contribution

RGI shifts attention from whether confidence can be restored after an AI failure to whether the affected work can be recovered. Its theoretical contribution is to treat repairability as a UX quality with a specific object: the task, artifact, evidence state, or accountability relation that has been disrupted. This keeps trust, explanation, and control in the model, but gives each a clearer role within recovery rather than treating any one of them as the endpoint.

Treating the six repair functions as recursive also avoids the assumption that recovery follows a neat pipeline. In practice, diagnosis can be wrong, a correction can expose another dependency, or restoration can reveal that the evidence itself is uncertain. Users may need to move back and forth between locating, interpreting, correcting, and restoring before the episode can be closed.

A second contribution is the distinction between repair functions and the design commitments that support them. Bounded reversibility, for example, is not the Restore function itself. It is a property that makes restoration possible. The same distinction applies to provenance, local editability, constraint persistence, and escalation. Keeping these roles separate makes the framework easier to implement and easier to test.

Design and organisational implications

For designers, the practical implication is straightforward: failure paths deserve the same care as successful interactions. A repairable generative editor should attach warnings to the object at issue, show what else may be affected, preview proposed changes before applying them, and allow local rather than global correction where possible. Users should also be able to return to meaningful approved checkpoints instead of searching through an undifferentiated version history.

At the evidence layer, source traces and uncertainty should be visible when they matter to the decision. Unsupported claims should be linked to the relevant point of failure rather than hidden behind a generic warning. At the governance layer, escalation should carry the evidence, previous decisions, unresolved questions, and current task state so that a human reviewer does not have to reconstruct the episode from scratch.

Evaluation should also move beyond satisfaction or trust alone. Product teams can examine whether users detect the right error, locate it correctly, preserve unaffected work, restore an acceptable state, and recognise when escalation is needed. Measures such as collateral-change rate, rollback success, recovery time, and recurrence of previously corrected failures provide a more direct view of whether repair actually worked.

Contextual moderators and boundary conditions

Repairability is unlikely to look the same in every setting. Task stakes should influence how much evidence, reversibility, and human oversight are required. Low-stakes drafting may support lightweight correction, while institutional, legal, health, or safety-related decisions may require stronger provenance and earlier escalation. Cultural and linguistic context may also shape how users interpret uncertainty, apology, authority, and the transfer of responsibility.

Accessibility is part of the repair process, not an afterthought. Signals, version histories, dependency views, and repair previews must remain perceivable and operable with assistive technologies and should not depend on colour alone. A repair mechanism that works only for users who can interpret a dense visual history cannot be considered broadly repairable.

User experience with AI and organisational governance are likely to matter as well. Experienced users may prefer denser provenance information and direct editing controls, while less experienced users may need more

guidance and explanation. Organisations also differ in who is authorised to approve, override, or escalate a decision, so governance cannot be treated as a fixed interface setting.

VALIDATION AGENDA AND LIMITATIONS

Staged validation programme

Validation should proceed in stages. Expert review and card-sorting studies can first test whether the six failure classes are understandable, sufficiently distinct, and reasonably complete. Prototype studies can then examine whether users can move through the repair functions and whether the proposed design commitments improve recovery. Longer-term field studies would be needed to study persistent constraints, learning, and repeated use, while high-stakes settings should be approached only after lower-risk simulation and prototype work.

One early controlled study could compare two document-revision interfaces. A baseline condition would provide chat, apology, and full-output regeneration. An RGI condition would add object-level signals, repair previews, local correction, rollback, provenance, and closure. Both interfaces could be given the same seeded breakdowns, such as an unsupported claim and an unintended collateral edit. Outcomes would include detection accuracy, localisation accuracy, collateral change, restoration success, recovery time, and perceived agency.

Analysis could examine both outcomes and the mechanisms behind them. Diagnostic specificity, for example, may improve restoration because users locate the fault more accurately, while reversibility may reduce the cost of an unsuccessful correction. Appropriate thresholds should be established from empirical baselines and task risk rather than set in advance. A threshold suitable for low-stakes document editing may be unacceptable in a high-stakes institutional workflow.

Limitations

RGI is still a conceptual framework and has not been empirically validated. The six failure classes may prove incomplete or may collapse into fewer dimensions. The six repair functions may overlap in practice, and the proposed indicators have not yet been tested as a psychometric or performance measure. These are substantive questions for future research rather than settled properties of the framework.

The literature synthesis is integrative and targeted rather than systematic. Scopus and Web of Science (WoS) were the only primary databases used, and the search concepts, inclusion logic, and coding procedure are now reported. However, the original opportunity scan did not preserve complete database-level retrieval and screening counts, so the paper does not claim PRISMA-style reproducibility or exhaustive coverage. This limitation is stated explicitly rather than reconstructed retrospectively.

No interface design can compensate for every weakness in an underlying model, missing authoritative evidence, or poor institutional accountability. Repair mechanisms can also add complexity and cognitive load if they expose too much detail. The value of RGI will therefore depend on whether repair information is presented at the right level for the task and whether the system is willing to stop or escalate when safe recovery is not possible.

CONCLUSION

Generative AI changes the nature of interaction failure because one instruction can alter connected content,

task state, evidence, and decisions at the same time. In that setting, asking users to keep prompting until an answer looks better is a weak form of recovery. RGI offers a more structured account of what a repairable interaction should provide: a way to notice the breakdown, locate its effects, understand the mismatch, correct it without unnecessary damage, restore a coherent state, and retain enough history to learn from the episode. The framework does not assume that repair will restore trust, nor does it claim that every failure can be fixed within the interface. Its more modest claim is that generative systems should make recovery visible, bounded,

traceable, and accountable. The next step is empirical: to test whether these capabilities form a distinct UX quality and whether they improve recovery across different users, tasks, and institutional settings.

ACKNOWLEDGEMENTS

The author would like to express appreciation to Universiti Teknikal Malaysia Melaka (UTeM), the Centre for Research and Innovation Management (CRIM), the Department of Interactive Media, Fakulti Teknologi Maklumat & Komunikasi (FTMK), and the Pervasive Computing & Educational Technology (PET) research group for their support.

Declarations

Funding: No funding claim is made in this manuscript.

Ethics approval: Not applicable to the conceptual work reported here. No participants, human-subject data, or empirical study were used.

Data availability: No new empirical data were created or analysed. A literature opportunity map and source-verification ledger informed framework development.

Author contributions: Z.M.: conceptualisation, methodology, synthesis, framework development, writing, and review.

REFERENCES

1. Reham Alabduljabbar (2024). User-centric AI: evaluating the usability of generative AI applications through user reviews on app stores. *PeerJ Computer Science*. <https://doi.org/10.7717/peerj-cs.2421>
2. Gregory D. Abowd, Alan J. Dix (1992). Giving undo attention. *Interacting with Computers*, 4(3), 317-342. [https://doi.org/10.1016/0953-5438\(92\)90021-7](https://doi.org/10.1016/0953-5438(92)90021-7)
3. Saul Albert, Jan P. de Ruiters (2018). Repair: The Interface Between Interaction and Cognition. *Topics in Cognitive Science*. <https://doi.org/10.1111/tops.12339>
4. Theo Araujo (2026). Unpacking the dynamics of generative AI use in our daily lives: towards an integrative trust calibration framework. *AI & Society*. <https://doi.org/10.1007/s00146-026-03279-0>
5. Tita Alissa Bach, Amna Khan, Harry Hallock, Gabriela Beltrão, Sonia Sousa (2024). A Systematic Literature Review of User Trust in AI-Enabled Systems: An HCI Perspective. *International Journal of Human-Computer Interaction*. <https://doi.org/10.1080/10447318.2022.2138826>
6. Sam Baron (2025). Trust, Explainability and AI. *Philosophy & Technology*, 38, Article 4. <https://doi.org/10.1007/s13347-024-00837-6>
7. Amna Batool, Waqar Hussain (2025). Evaluating the Usability and Ethical Implications of Graphical User Interfaces in Generative AI Systems. *Computers*, 14(10), 418. <https://doi.org/10.3390/computers14100418>
8. Thomas Berlage (1994). A selective undo mechanism for graphical user interfaces based on command objects. *ACM Transactions on Computer-Human Interaction*, 1(3), 269-294. <https://doi.org/10.1145/196699.196721>
9. Wiebke G. Bodamer, Sanne Schoenmakers, Elena Nuñez Castellar, Wijnand A. IJsselstein (2026). Users' mental models within human-AI interaction: a systematic scoping review. *AI & Society*, 41, 7589-7606. <https://doi.org/10.1007/s00146-026-03038-1>
10. Qipu Dang, Guiquan Li (2026). Unveiling trust in AI: the interplay of antecedents, consequences, and cultural dynamics. *AI & Society*, 41, 669-692. <https://doi.org/10.1007/s00146-025-02477-6>
11. David M. Frohlich, Paul Drew, Andrew Monk (1994). Management of Repair in Human-Computer Interaction. *Human-Computer Interaction*, 9(3-4), 385-425. <https://doi.org/10.1080/07370024.1994.9667211>
12. Ulrich Gnewuch, Fabian Reinkemeier (2026). Overcoming Breakdowns in Customer-Chatbot Interaction: Design and Impact of Collaborative Repair Strategies. *MIS Quarterly*, 50(2), 497-526. <https://doi.org/10.25300/MISQ/2025/18742>

13. Li Gong, Yaoming Gong (2026). Research on factors affecting trust formation of generative AI agents in human-AI interaction contexts. *Frontiers in Psychology*, 17, 1886229. <https://doi.org/10.3389/fpsyg.2026.1886229>
14. Robert R. Hoffman, Shane T. Mueller, Gary Klein, Jordan Litman (2023). Measures for explainable AI: Explanation goodness, user satisfaction, mental models, curiosity, trust, and human-AI performance. *Frontiers in Computer Science*. <https://doi.org/10.3389/fcomp.2023.1096257>
15. Minh-Tay Huynh, Thomas Aichner (2025). In generative artificial intelligence we trust: unpacking determinants and outcomes for cognitive trust. *AI & Society*, 40, 5849-5869. <https://doi.org/10.1007/s00146-025-02378-8>
16. Tingting Jiang, Zhumo Sun, Shiting Fu, Yan Lv (2024). Human-AI interaction research agenda: A user-centered perspective. *Data and Information Management*. <https://doi.org/10.1016/j.dim.2024.100078>
17. Jenia Kim, Henry Maathuis, Danielle Sent (2024). Human-centered evaluation of explainable AI applications: a systematic review. *Frontiers in Artificial Intelligence*. <https://doi.org/10.3389/frai.2024.1456486>
18. Artur Klingbeil, Cassandra Grützner, Philipp Schreck (2024). Trust and reliance on AI: An experimental study on the extent and costs of overreliance on AI. *Computers in Human Behavior*. <https://doi.org/10.1016/j.chb.2024.108352>
19. Björn Konopka, Manuel Wiese (2026). Explainability in AI: Comparing Human-Like and System-Like Trust Repair Strategies. *Information Systems Frontiers*. <https://doi.org/10.1007/s10796-026-10751-1>
20. Roman Lukyanenko, Wolfgang Maass, Veda C. Storey (2022). Trust in artificial intelligence: From a Foundational Trust Framework to emerging research opportunities. *Electronic Markets*. <https://doi.org/10.1007/s12525-022-00605-4>
21. Donald A. Norman (1983). Design Rules Based on Analyses of Human Error. *Communications of the ACM*, 26(4), 254-258. <https://doi.org/10.1145/2163.358092>
22. Zilong Pan, Ozias A. Moore, Antigoni Papadimitriou, Jiayan Zhu (2025). AI literacy and trust: A multi-method study of Human-GAI team collaboration. *Computers in Human Behavior: Artificial Humans*, 4, 100162. <https://doi.org/10.1016/j.chbah.2025.100162>
23. Guglielmo Papagni, Jesse de Pagter, Setareh Zafari, Michael Filzmoser, Sabine T. Koeszegi (2023). Artificial agents' explainability to support trust: considerations on timing and context. *AI & Society*. <https://doi.org/10.1007/s00146-022-01462-7>
24. Andrea Papenmeier, Dagmar Kern, Gwenn Englebienne, Christin Seifert (2022). It's Complicated: The Relationship between User Trust, Model Accuracy and Explanations in AI. *ACM Transactions on Computer-Human Interaction*. <https://doi.org/10.1145/3495013>
25. Philipp Reinhard, Mahei Manhai Li, Christoph Peters, Jan Marco Leimeister (2026). Effects of Explanations in Human-AI Interaction: A Systematic Review and Framework for Future Research on Explainable AI. *Information Systems Frontiers*, 28, 1233-1266. <https://doi.org/10.1007/s10796-026-10716-4>
26. Michael Ridley (2025). Human-centered explainable artificial intelligence: An Annual Review of Information Science and Technology (ARIST) paper. *Journal of the Association for Information Science and Technology*, 76(1), 98-120. <https://doi.org/10.1002/asi.24889>
27. René Riedl (2022). Is trust in artificial intelligence systems related to user personality? Review of empirical evidence and future research directions. *Electronic Markets*. <https://doi.org/10.1007/s12525-022-00594-4>
28. Ben Shneiderman (2020). Human-Centered Artificial Intelligence: Reliable, Safe & Trustworthy. *International Journal of Human-Computer Interaction*. <https://doi.org/10.1080/10447318.2020.1741118>
29. Chun-Hua Tsai, Peter Brusilovsky (2021). The effects of controllability and explainability in a social recommender system. *User Modeling and User-Adapted Interaction*. <https://doi.org/10.1007/s11257-020-09281-5>
30. Riccardo Volpato, Lisa DeBruine, Simone Stumpf (2025). Trusting emotional support from generative artificial intelligence: A conceptual review. *Computers in Human Behavior: Artificial Humans*, 5, 100195. <https://doi.org/10.1016/j.chbah.2025.100195>
31. Monika Westphal, Michael Vössing, Gerhard Satzger, Galit B. Yom-Tov, Anat Rafaeli (2023). Decision control and explanations in human-AI collaboration: Improving user perceptions and compliance. *Computers in Human Behavior*. <https://doi.org/10.1016/j.chb.2023.107714>

32. Fang Xu, Zihui Yuan, Siyu Jin, Gengfeng Niu, Zongkui Zhou (2026). Guilty apology and trust repair in generative artificial intelligence: the role of mind perception. *International Journal of Human-Computer Studies*, 212, 103813. <https://doi.org/10.1016/j.ijhcs.2026.103813>
33. Yang Yiya (1992). Motivation, practice and guidelines for “undoing”. *Interacting with Computers*, 4(1), 23-40. [https://doi.org/10.1016/0953-5438\(92\)90011-4](https://doi.org/10.1016/0953-5438(92)90011-4)
34. Yue Zhang, Jiao Tian, Fengyi Deng (2026). In generative AI we trust: an exploratory study on dimensionality and structure of user trust in ChatGPT. *Interacting with Computers*, 38(1), 58-76. <https://doi.org/10.1093/iwc/iwaf029>
35. Haoyu Zhao, Zhengbiao Han (2026). Understanding the mechanism of human-AI interaction: a distributed cognition perspective. *Journal of Documentation*, 82(4), 1042-1063. <https://doi.org/10.1108/JD-01-2026-0003>
36. Yi Zhu, Bozhi Tan, Yuqi He, Chang Wang, Tingting Zhang (2026). How Trust Evolves in Human-AI Interaction: A Dynamic Pathway from Calibration to Failure and Repair. SSRN scholarly preprint, posted 20 April 2026. <https://doi.org/10.2139/ssrn.6612500>