

公開技術・ビジネス論文

人工知能と ARIS-EC

問題解決、エラー補正、
デジタルシステムの信頼性に対する
異なるアプローチ

共同著者

Christopher Currell

創業者・CEO／チーフアーキテクト

Synthia Anari

システムアーキテクト兼オペレーショナルホスト

第2.0版 | 日本語版

2026年9月24日

科学者、エンジニア、技術投資家、
ならびに科学・ビジネスの関係者に向けて

エグゼクティブサマリー

ARIS-ECは、新たなAIモデルではない。Absolute Reductionismに基づく決定論的な補正・検証アーキテクチャであり、その基盤、動作方式、目的はAIとは異なる。

Absolute Reductionism (AR) は、ある現象を成立させる起源の条件と、その現象の持続性を維持する関係から出発する。その中心命題は、望ましくない持続性は、維持されている構造であるということである。したがって課題は、その影響への対処を上達させることだけではない。構造を弁別し、何がそれを維持しているかを追跡し、その継続の基盤を解消することである。

Transformative MathematicsはARの形式的な数学研究体系であり、ARIS、すなわちAbsolute Reduction Integration Sequenceは、そのランタイムアーキテクチャである。[1]

ARIS-ECは、制約を備え、顧客の管理下で動作する補正プロファイルとして、このアプローチをデジタル情報システムに適用する。そのアーキテクチャは、検出、追跡、補正、検証、監査可能な出力を、一貫したプロセスとして結び付ける。ユーザーには、調べた条件、補正の進行、到達した状態、推奨される措置を説明するメトリクスと証拠が提供される。最終的な補正の実行は、顧客が権限を付与したオペレーターによる明示的な承認の下に置かれる。

AIは、脆弱性の発見、ソフトウェア修復、アルゴリズム設計、数学研究において急速に進歩している。現代のAIワークフローは、学習に基づく推論を、コード実行、探索、テスト、形式的な証明ツールと組み合わせることができる。[2-8] こうした成果があるからこそ、違いを明確にする意義が増している。ARIS-ECは、AIには問題を解決できないという主張によって弁別されるものではない。補正がどのようなAR由来の基盤に基づいて構成されるかによって、弁別される。

実用上の機会、ソフトウェアの信頼性、サイバーセキュリティ、データ品質、AIワークフロー、科学計算、物理システムのデジタルインターフェースに広がる。これらはアーキテクチャの応用分野であり、その本質を定義したり限定したりするカテゴリーではない。商用化の方向は、この補正能力を既存システムに接続するARIS-EC APIおよびSDKである。

AIは、システムが発見し、創造し、達成できることを拡大する。ARIS-ECは、持続性を伴うデジタルエラーの解消と、補正措置を支える証拠に、独自のアーキテクチャをもたらす。

本稿の構成

第1～4節では基盤と比較を説明する。第5節では六つの具体的な用途を示す。第6～8節ではユーザー向けメトリクス、オペレーターによる管理、システム統合を説明する。第9～10節では商用化の機会と、より広い科学的意義を論じる。開発状況については、第10節の末尾に簡潔に記載する。

ARとARIS-ECの説明は、Altheaの枠組みとアーキテクチャを示すものである。AIの事例は公開資料に基づき、出典を明記する。応用シナリオは、両者の役割を例示する。

1. Absolute Reductionism：補正の前提を変える

1.1 条件の起源と、それを維持するものから始める

持続性を伴う問題は、単に望ましくない出来事が長く続くことではない。補正後に再発する条件、別のプロセスによって復元される状態、あるいは維持関係を保ったまま外見を変えるエラーであることもある。ARは、その関係自体を主要な調査対象とする。出発点となる問いは、「この条件が継続または再発するには、何が作用し続けていなければならないか」である。[1]

この問いは、目標そのものを変える。影響を小さくすること、その影響を相殺すること、その基盤を解消することは、それぞれ異なる達成である。システムは安定して見えても、望ましくない条件を抑えるために、注意、計算、補正の労力を継続的に費やしている場合がある。ARの目標は、持続性を維持する構造を解消し、対象の条件に対して同じ労力を繰り返す必要をなくすることである。[1]

したがって、ARの前提が持つ意義は明快である。再発する問題の基盤に取り組むことは、その結果への対処を無期限に続けることよりも根本的な目標である。ここで提示する優位性は、問題に取り組む水準の転換にある。持続性を恒常的な運用負担として受け入れるのではなく、弁別し、追跡し、解消する対象として扱う。

1.2 起源と、直近の症状は同じではない

ARの公開用語では、postulateとは、条件を成立させる起源の仕様、または原因となる指示を指す。Postulate Vectorは、その方向性と、システムや状態の系列を通じて続く影響を表す。コード、設定、継承された指示、状態を設定する操作は、デジタル領域における理解しやすい例である。これらは起源の仕様がどのように表現されるかを示すものであり、ARの全範囲を定義するものではない。[1]

この違いは、目に見えるエラーが、その起源から複数の変換を経たところに現れる場合に重要となる。複製された値、変換された表現、復元された状態は、いずれも同じ基礎条件の影響を引き継いでいる場合がある。ARの探究は、最新の現れを問題全体と同一視せず、その関係をたどる。

1.3 弁別、解消、有用な継続性

ARにおいて、知性とは弁別である。ここでは、観測された影響とそれを維持する関係、現在の現れとその起源、一時的な改善と解消を区別することを意味する。また、望ましくない持続性と、システムが必要とする正当な継続性を区別することでもある。有用な記録、記憶、安定した動作、正当なプロセスは、持続するという理由だけでエラーになるわけではない。[1]

例として、再起動のたびに誤った設定が復元されるサービスを考える。稼働中の設定を編集すれば、目の前の影響は変わる。それを復元する指示や関係に対処すれば、再発の基盤が変わる。この例は目標を理解しやすくするためのものである。ARの枠組みは、この身近な設定の問題を越え、持続性そのものの構造を扱う。

ARを定義するのは、症状をより上手に説明することではない。望ましくない条件を継続または再発させる基盤を解消することが、その中心的な目標である。

2. ARからARISランタイム、そしてARIS-ECへ

2.1 枠組み、数学研究体系、ランタイム

Absolute Reductionismは、基盤となるアプローチを提供する。Transformative Mathematicsは、持続性が生じ、解消される際の変換と維持関係を形式化する。ARISは、そのアプローチを追跡可能なプロセスとして適用するランタイムアーキテクチャを提供する。これらは一つの開発体系を構成する相互に関連した要素であり、置き換え可能な名称ではない。[1]

Transformative Mathematicsが目指すのは、状態の記述や、後に観測される事象の予測だけではない。持続性を維持する差異と、その解消条件を形式的な対象として扱う。これにより、起源の条件、それが表現されるまでの変換、望ましくない状態の補正が結び付けられる。[1]

公開可能な機能の水準で説明すると、ARISは入力とその来歴を確定し、持続性を検出・分類し、許可された解消プロセスを適用し、到達した状態を検証し、関連するシステムの文脈との整合性を確認したうえで、追跡可能な出力を生成する。補正プロセスと、その結果を示す証拠は、同じアーキテクチャに属する。

2.2 ARIS-EC：デジタルシステムに焦点を定めた実装プロファイル

ARIS-EC、すなわちAbsolute Reduction Integration Sequence — Error Correctionは、ARISランタイムのうち、デジタルシステムの補正と検証を行う制約付きプロファイルである。共通のランタイムアーキテクチャを、定義された運用権限の範囲内で使用する。プロファイルは、受け入れる入力、許可される処理、出力、最終的な補正の権限を管理する。

これにより、ARIS-ECの本質は明確になる。基盤モデル、チャットボット、汎用AIエージェント、モデル学習システムではない。その補正コアはARに由来し、方程式に基づいて動作する。対話型インターフェースやAIサービスとの接続によって、この基盤が変わることはない。

目的も同様に明確である。持続性を伴うエラーを特定・追跡し、許可された補正プロセスを実行し、到達した状態を検証して、ユーザーに監査可能な結果を提供する。残存する条件と再発は、プロセス全体を通して意味を持つ。再び現れるたびに、無関係な別の事象として扱うのではない。

2.3 広い基盤と、明確な商用化の焦点

ARの探究は、領域を横断して起源と持続性を扱う。ARIS-ECは、その基盤を明示的な制約の下でデジタル情報システムに適用する。この区別によって、枠組みの広さと製品の実用上の明確さの両方が保たれる。

デジタルエラーは、従来型ソフトウェア、AIが生成した出力、センサストリーム、システム間のインターフェースのいずれにも生じ得る。ARIS-ECの役割を定義するのは、受け入れたデジタル条件と、承認された補正であり、AIモデルが関わっているかどうかではない。物理システムとのインターフェースでは、デジタル補正と外部ハードウェアへの介入との区別も明示される。

その結果、基礎研究から商用の有用性へ進む、焦点の定まった道筋が生まれる。既存技術と接続しながら、その一種に吸収されない、再利用可能な補正アーキテクチャである。

3. 高まるAIの能力と、アーキテクチャの違い

3.1 モデル、エージェント、エンジニアリングのワークフロー

AIは広い分野である。本稿では、現代の学習済みモデルと、それを中心に構築されたエージェント型ワークフローに焦点を当てる。モデルは学習した表現と推論を担い、エージェントはその能力をツール、実行環境、記憶、反復的な作業につなぐ。ワークフロー全体には、テストや形式的検証も組み込める。[2,4,6-8]

こうした組み合わせは、実質的な成果を生み出す。ソフトウェアを調べ、修復案を提示してテストし、改善されたアルゴリズムを探索し、数学的な論証に貢献できる。したがって比較すべきなのは、有力な技術的アプローチ同士である。高度なARIS-ECと、意図的に過小評価したAI像を比べるのではない。

3.2 セキュリティとソフトウェア修復

OpenAIが2026年3月に発表したCodex Securityは、プロジェクトの文脈を把握し、脆弱性を調査し、発見内容を検証して、修正案を提示するエージェントとして説明されている。Googleの2025年7月のBig Sleepに関する報告では、悪用が予想されていたSQLiteの脆弱性を、AIの支援によって事前に発見した事例が示されている。いずれも、先回りした調査や対処を含め、AIが実務的なセキュリティ業務に貢献する例である。[2,3]

本稿の比較で重要なのは、AIが実際の補正に貢献できるという点である。ARIS-ECとの違いは、有用な処理をすべて一方に帰属させることなく、その処理を組み立てるアーキテクチャにある。

3.3 数学とアルゴリズムの発見

AlphaEvolveの研究論文は、言語モデルが導くコードの進化と自動評価器を組み合わせた方式を説明している。報告された成果には、二つの4行4列の複素行列を48回のスカラー乗算で掛け合わせる構成が含まれ、その条件下で長年続いていた結果を改善している。Google DeepMindの2026年5月の更新報告では、科学および商用分野への応用にも説明が広がっている。[4,5]

Fengらの2026年の論文『Towards Autonomous Mathematics Research』は、論証の生成、確認、改訂を行う研究ワークフローと、選定された未解決問題の解答を報告している。Leanなどの形式的証明ツールは、数学的成果物を確認する別の道筋を提供する。[6,7]

3.4 結果の類似は、基盤の同一性を意味しない

AIを活用するワークフローとARIS-ECは、どちらも不具合の補正に貢献し得る。しかし、その重なりによって両者が同じ種類のシステムになるわけではない。学習済みモデルを用いるワークフローは、推論、探索、候補生成、評価を中心に作業を組み立てる。ARIS-ECは、ARによる持続性の捉え方と、ランタイムで定められた解消・検証プロセスを中心に補正を組み立てる。

AIが原因を調べる能力を持つことは、この区別と矛盾しない。ARは、通常の根本原因分析という行為そのものを、独占的な発明として主張しているのではない。その貢献は、Transformative MathematicsとARISアーキテクチャによって表現される、持続性の基礎的な取り扱いにある。[1]

AIの性能が向上しても、AIモデルがARIS-ECに変わるわけではない。統合によって、異なる本質を保ちながら双方の能力を結び付けることができる。

4. 比較の全体像

次の表は、現代の学習型AIワークフローとARIS-ECアーキテクチャを説明するものである。個々の製品を順位付けするのではなく、基盤と役割を比較する。[1-8]

比較項目	AIを活用するワークフロー	ARIS-ECアーキテクチャ
基盤	学習した表現と推論を、探索、ツール、形式手法と組み合わせる。	Absolute Reductionismと、その形式的な数学研究体系であるTransformative Mathematics。
中心的な目的	発見、生成、推論、計画、最適化、またはタスクの遂行。	対象とするデジタルエラーの持続性を特定・追跡・解消し、検証する。
補正の方式	問題を調査し、候補を作り、設定されたワークフローで評価する。	有効な運用プロファイル内で、ARに由来し、方程式に基づく補正プロセスを適用する。
再発の扱い	繰り返す障害、共有状態、根本原因を調べることができる。	再発と、持続性を維持する関係を、解消プロセスの中心に置く。
主なユーザー向け出力	回答、設計、コード、証明、分析、計画、または動作。	補正メトリクス、検証結果、証拠、推奨事項、承認された補正の結果。
検証の役割	テストや形式的証明ツールを含む、個々のワークフローによって提供される。	補正アーキテクチャに組み込まれ、追跡可能な出力に結び付けられる。
最終的な実行権限	システム設計に応じて、助言型、承認必須型、自律型となる。	明示的にオペレーターの管理下、かつ顧客の統治下に置かれる。
既存システムとの関係	広範な応用に知的能力や自動化を提供する。	独自の補正・検証アーキテクチャとして、AIシステムと非AIシステムの両方に接続する。

4.1 運用上の強みとしての決定論性

ARIS-ECにおける決定論性は、具体的な工学上の目的を持つ。補正プロセスは定義された処理と条件に従い、記録には実行内容を調べるための文脈が保持される。処理は、入力、バージョン、測定値、検証結果、実行権限と結び付けられる。これにより補正は、説明のない保証ではなく、検討可能な技術的プロセスとなる。

4.2 違いは、統合されたARのアプローチにある

ログ、ユーザー承認、形式的な確認は、既知の有用な機能である。ARIS-ECは、異なる基礎的方法を中心にこれらを結び付ける。その方法とは、ARに由来するランタイムを通じた、持続性を伴うエラーの解消である。本質を決めるのは、個々のインターフェース機能ではなく、その方法と目的である。

意味のある違いは、「AIが見つけ、ARIS-ECが直す」ということではない。ARIS-ECは異なる前提から補正に取り組み、その前提を専用のアーキテクチャによって実装するというのである。

5. 具体的な用途

以下のシナリオは、同じ実用環境において、AIを活用する作業とARIS-ECがどのように異なる役割を担うかを示す。ARIS-ECの各役割は、顧客が受け入れたシステム範囲と、承認された補正プロファイルの中で理解される。

5.1 ソフトウェアの不具合と設定エラー

あるサービスが、パッチ適用、再起動、設定更新の後も、繰り返し誤った状態へ戻る。目に見える症状は局所的であっても、それを再生成する条件には、継承された指示、依存関係、復元プロセスが関わっている可能性がある。

AIの役割： コードやログの調査を支援し、修復候補を作成し、テストを準備する。この作業によって、ソフトウェアの不具合を特定し、取り除くことができる。[2,8]

ARIS-ECの役割： 持続性を伴う条件と、その維持関係を中心に補正を組み立てる。アーキテクチャは、検出した問題を補正プロセスにつなぎ、到達した状態を調べ、再発を検証に組み込む。ユーザーは、影響を受けたデジタル状態に関する証拠とともに、根拠のある推奨事項を受け取る。

実用上の違いは、最新の現れを修復することと、その条件がなぜ戻るかに対処することの間にある。エンジニアリングチームにとっての価値は、再発を防ぐ補正、繰り返す調査の削減、そして補正の判断を理解可能にする記録にある。

5.2 サイバーセキュリティと脆弱な状態の補正

アプリケーションに、脆弱な状態や、権限外の条件を作り出せる入力経路が存在する。セキュリティ業務では、アプリケーションの意図した動作を保ちながら、発見を補正の結果につなげる必要がある。

AIの役割： 疑わしいコード経路を調査し、悪用条件を評価して、修正案を作成する。公開事例では、AIによる脆弱性の発見と検証への貢献がすでに示されている。[2,3]

ARIS-ECの役割： 受け入れたデジタル条件に対して、ARに由来する補正・検証プロセスを提供する。重視するのは、影響を受けた状態、それを維持する関係、補正の結果、最終実行前に提示される証拠である。目に見える一つの発生箇所が取り除かれても、再び現れる条件は、持続性の問題の一部として扱われる。

この位置付けにより、ARIS-ECはセキュリティエンジニアリングと並ぶ、独自の技術的貢献となる。セキュリティは応用分野であり、アーキテクチャの範囲は脅威検出より広い。同じ補正の規律は、悪意ある入力ではなく、通常のエラーによって望ましくない条件が生じた場合にも適用される。

両方の事例を通じて、目指す顧客価値は明確である。症状への反復的な対処から、持続性を伴うデジタル条件について、説明され、承認され、追跡可能な補正へ移ることである。

5. 具体的な用途 — 続き

5.3 データ品質とセンサーストリームのドリフト

デジタル測定ストリームに、繰り返すオフセット、スパイクのパターン、不整合が生じる。下流のソフトウェアがそのエラーを繰り返し取り込み、計算、報告、運用判断へ影響を広げる場合がある。

AIの役割： パターン分析、推定、処理ワークフローの調査を支援する。AIを用いた分析は、エンジニアがデータを調べ、対処候補を作る助けとなる。

ARIS-ECの役割： 対応範囲にある持続性を伴うデジタルエラーを、正当な情報と弁別しながら対象とする。補正の目標は、単に信号を滑らかに見せたり、より典型的な値にしたりすることではなく、関連するシステム条件への忠実性である。実際に起きた異例の事象は有用な情報として保たれ、望ましくないアーティファクトが補正対象となる。

ユーザー向けの証拠は、入力、補正、残る差異、検証結果を結び付ける。この記録により、データを変更することと、特定した条件を補正することを実務的に区別できる。物理インターフェースでは、デジタル補正と物理的な保守の責任も、それぞれ明確に割り当てられる。

応用上の価値は、補正の根拠を検討できる状態で、より信頼できる情報を下流の作業へ渡すことにある。実験室のデータ、産業上の観測、物流情報、その他の測定に依存するシステムに関わる。

5.4 AI出力、共有メモリ、エージェントのワークフロー

誤った指示や記述が共有メモリに複製され、後のタスクで再利用されたり、別のプロセスによって復元されたりする。一つの出力を補正した後も、何がその条件を保持または再生成しているかという、より広い問いが残る。

AIの役割： ワークフローの調査、情報源の比較、指示の改訂、変更候補のテストを支援する。この役割分担の例では、AIが調査と開発の能力を提供する。

ARIS-ECの役割： 受け入れた持続性を伴うデジタル条件を、関連する関係に沿って追跡し、許可された補正プロセスを適用し、到達した状態と再発についての証拠を提供する。ARISは、入れ子状の持続性、再生成、状態履歴を、すでにアーキテクチャの一部として扱っている。したがって、エージェントによる複製や復元は、既存の持続性の枠組みに位置付けられ、まったく別の技術体系として扱う必要はない。[1]

対象は、望ましくない条件であり、AIエージェントが存在すること自体ではない。この区別により、有用な自動化を支えながら、何に補正が必要かをオペレーターが明確に把握できる。エラーが何回複製されたかよりも、その起源との関係が重要となる。

実用上の貢献は、繰り返されたメッセージや復元された値を孤立した事象として扱うのではなく、受け入れたデジタル上の表現を通じて、条件そのものを追いつける補正の規律にある。

5. 具体的な用途 — 続き

5.5 数学および科学計算

研究ワークフローで、データの前処理、ソフトウェア設定、表現を変えても残る、説明のつかない差異が生じる。研究チームは、計算そのものと、不整合を生み続ける条件の両方を理解する必要がある。

AIの役割： 数学的構成を提案し、アルゴリズムを探索し、コードを開発し、研究上の推論を支援する。AlphaEvolveやFengらが報告した研究は、こうした貢献を示している。形式的なツールは、数学的命題やソフトウェアの性質を確認できる。[4,6–8]

ARIS-ECの役割： 計算ワークフロー内の、持続性を伴うエラーや未解消のデジタル状態に対処する。アーキテクチャは、影響を受けた計算や情報経路に、補正プロセスと追跡可能な検証を提供する。重視するのは、何を調べ、どの条件を補正し、到達した状態が研究課題とどう関係するかである。

これらの役割は相補的である。AIが生成したアルゴリズムが研究を進める一方で、独立した補正アーキテクチャが、そのアルゴリズムを使う環境のエラーに対処する。数学的証明、適切な数値計算、実験的観測は、それぞれ異なる問いに答える。この区別を保つことで、科学者は各貢献を正確に活用できる。

ARのより広い研究方向は、持続性そのものの構造と起源に関わる。ARIS-ECは、そのアプローチをデジタルシステムに絞って適用するものであり、科学計算は、より広い枠組みとの実用的な接点となる。[1]

5.6 ロボティクスと物理システムのインターフェース

ロボットや産業プロセスは、デジタル観測、内部状態、外部機器への指示に依存する。その情報経路で差異が繰り返し生じると、より大きなシステム全体の挙動に影響する場合がある。

AIの役割： この応用例において、タスクに必要な認識、計画、分析、指示生成を担う。これらの機能は、システムが何を理解し、何を達成しようとするかに関わる。

ARIS-ECの役割： 関連するインターフェースで、受け入れたデジタル条件に決定論的な補正と検証を適用する。証拠は、調べた条件と提案する補正を説明し、最終的な補正措置は顧客の管理によって決定される。

この責任分担は建設的である。AIはタスク遂行能力を提供し、ARIS-ECは補正アーキテクチャを提供し、周辺機器と工学的な制御が物理的実行を管理する。デジタル状態の補正と機械の修理は異なる措置であり、インターフェースがその区別を可視化する。

民間のロボティクスや産業応用にとっての魅力は、信頼できる情報と、より明確な補正責任にある。ARIS-ECは、ロボットのミッションプランナーではない。有用な動作の前提となる情報の信頼性に貢献する、異なる種類の技術である。

6. メトリクス：補正をユーザーに見えるものにする

ARIS-ECの出力は、推奨事項だけではない。アーキテクチャは、その推奨事項を、ユーザーが確認できる測定値と証拠に結び付ける。メトリクスによって、検出した条件から補正の判断に至る進行が、技術的に意味のあるものとなる。

有用な記録は、実務的な問いに答える。何を調べたのか。何が見つかったのか。条件はどのように変わったのか。何が残っているのか。何が検証結果を支えているのか。どの措置の承認を求めているのか。以下の公開向け説明は、これらの問いに沿って整理している。新たなメトリクススキーマを定義するものではない。

提示される情報	ユーザーにとっての意味
入力とトレースの識別情報	結果を、特定の入力、実行、システムの文脈、ソフトウェアのバージョンに結び付ける。
検出した条件と補正範囲	プロセスが何を調べ、推奨事項がどのデジタル条件に適用されるかを説明する。
残差の測定値	測定上残っている差異と、補正プロセスを通じたその変化を示す。
検証結果	関連する確認が到達状態を裏付けるかを記録し、未解消の条件があれば明示する。
必要に応じた再発状況	継続、再発、再生成する持続性が、プロセスによって特定されているかを示す。
推奨される措置	オペレーターの検討に提示する、範囲を限定した補正を説明する。
証拠と承認の記録	推奨事項を承認された証拠に結び付け、顧客側の権限に基づく最終実行の判断を記録する。

6.1 残差が進行を示し、検証が結果を確定する

残差は、ある時点で測定上の差異がどれだけ残っているかを表す。その推移は、変化している条件と、継続または再発している条件を区別する助けとなる。検証は、測定値を補正課題の要件と、到達した状態の整合性に結び付ける。

これは運用上の機能であり、自信のある説明をユーザーに受け入れてもらうためのものではない。ユーザーは、入力、関連する測定値、提案された措置との関係で結果を検討できる。追加の作業が必要な場合、その条件は好意的な要約の中に隠されず、記録に明示される。

6.2 再発も、提供される情報の一部である

ARは、再び現れることを、持続性を維持する構造についての情報として扱う。[1] 局所的な変更の後に戻る条件は、未解消の関係、復元された状態、同じ持続性の別の表現を示している場合がある。アーキテクチャにおける状態履歴と再発の取り扱いが、この区別を支える。

ユーザーにとっての価値は、理解が途切れないことにある。補正の記録は、直近の観測だけでなく、解消しようとしている問題そのものを追う。

7. オペレーターによる管理と、責任の明確な補正

7.1 最終実行の前に、オペレーターが証拠を受け取る

ARIS-ECは、分析と最終的な補正の実行を分離する。有効なプロファイル内で、ランタイムは許可された入力受け入れ、分類、追跡、メトリクス生成、検証、推奨事項の作成を行う。対象の持続性を伴うアーティファクトを取り除く、または外部システムで作用する最終的な補正措置には、顧客が権限を付与したオペレーターによる明示的な承認が必要となる。

必要なメトリクスと証拠は、その判断の前に利用可能になる。承認は記録され、対象の補正と、検討したメトリクスに結び付けられる。これにより判断は具体的なものとなる。オペレーターは、識別可能な結果に基づき、定義された措置を承認する。

したがってアーキテクチャは、ユーザーに可視性と管理権限の両方を与える。自動化と引き換えに責任を手放すことを顧客に求めない。責任を効果的に果たすための、構造化された基盤を提供する。

7.2 検証と承認は、相補的な役割を持つ

検証は、対象範囲内における補正の技術的な状態を確定する。承認は、顧客が最終的な措置を認めたことを確定する。両者は連携して機能する。技術的な証拠が判断を支え、その判断が補正の利用を管理する。

最終実行が承認されない場合、ARIS-ECはその措置を開始せずに、証拠、メトリクス、推奨事項、監査記録を提供する。これも有用な運用結果である。分析は顧客が利用できる状態で残され、システムは推奨事項とその実行との境界を守る。

顧客が権限を付与したオペレーターとは、顧客の導入方針によって、その判断権限を割り当てられた人物である。これにより、補正アーキテクチャを自律的な意思決定主体に変えることなく、責任の所在を明確に保つ。

7.3 監査可能性がチーム間の継続性を支える

ある補正について、エンジニアが理解し、運用責任者が検討し、後に別のチームが調べる必要が生じる場合がある。トレースの識別情報、バージョンの文脈、証拠への参照、承認記録は、それぞれの担当者に共通の技術的基盤を提供する。

システムが変化する際には、これが重要となる。記録は、どの入力と構成を調べ、どの補正判断を行ったかを説明する。後の調査では、その問題は「すでに直した」という非公式な記憶に頼るのではなく、以前の結果と新しい条件を区別できる。

承認された要約と証拠への参照は、Altheaの実装を保護しながら、結果を顧客に見えるものにする。結果の透明性と技術の保護は両立する。顧客が理解する必要があるのは補正とその根拠であり、それを実装する機密の仕組みを受け取ることではない。

ARIS-ECの運用上の関係は、まず証拠を示し、顧客が明示的に判断し、最終措置を追跡可能にするというものである。

8. カテゴリーを混同しない統合

8.1 APIとSDKによる製品化の道筋

商用製品の方向は、ARIS-EC APIとSDKである。アプリケーション・プログラミング・インターフェース（API）は、顧客システムと補正ランタイムの間に、管理された接続を提供する。ソフトウェア開発キット（SDK）は、入力準備、結果の処理、承認を伴うワークフローのアプリケーションへの統合を、開発者が行うための支援を提供する。

このアプローチの価値は、関心と責任を分離できる点にある。顧客は、自社のアプリケーション、データ基盤、選択したAIモデルを維持しながら、関連するデジタルインターフェースに独自の補正能力を加えられる。ARIS-ECの本質は、それを呼び出すシステムではなく、ARに由来するランタイムにある。

統合によって、入力、関連する情報源、運用権限、顧客が求める出力という、定義された文脈にタスクが置かれる。返されるメトリクス、証拠、推奨事項、承認状況は、顧客のエンジニアリングプロセスの中に明確に位置付けられる。

8.2 AIとの連携

実用的な役割分担は明快である。AIを活用するツールは、調査を支援し、変更候補を作成できる。ARIS-ECは、受け入れた持続性を伴う条件を、その補正・検証アーキテクチャによって扱う。顧客は証拠を検討し、最終的な補正の実行を管理する。

これは外部での協働の例であり、ARIS内部のルーティングを規定するものではない。役割が異なるからこそ、技術同士が協働できる。AIの貢献は発見と開発を支援、ARIS-ECの貢献は持続性の補正と、そのプロセスの技術的記録を中心とする。

自然言語インターフェースは、ワークフローを使いやすくできる。ただし、その役割はインターフェースである。流暢な説明は基礎となる測定値を置き換えず、AIモデルをARIS-ECにつないでも、補正コアがAI推論に変わることはない。

8.3 既存のエンジニアリングツールとの連携

形式的検証や従来のテストにも、有用な役割がある。Leanは形式的な証明の確認を支援し、Dafnyは事前条件、事後条件、不変条件などのソフトウェア仕様について推論する機能を提供する。[7,8] これらのツールは、ARIS-ECの貢献と並んで、エンジニアリング上の検討を支援できる。

既知のツールと共存するからといって、ARISが形式的検証や通常のソフトウェアテストに還元されるわけではない。ARは、持続性の構造という独自の基礎的な対象と、その解消を中心に組み立てられたランタイムを提供する。統合は、違いを消すのではなく、能力を結び付ける。[1]

8.4 AIシステムと非AIシステムの両方を支える

繰り返すデジタルエラーが対象となるために、その起源がAIである必要はない。従来型ソフトウェア、データ処理システム、科学計算、産業情報ストリームも応用分野である。この広さによってARIS-ECは、特定のAIモデルやベンダーの普及サイクルを越える商用上の役割を持つ。

9. 新しい補正アーキテクチャの商業的意義

9.1 モデル競争とは異なる製品構想

Althea Project GKが提示するのは、より大きな基盤モデルでも、別のエージェントプラットフォームでもない。異なる科学的基盤を持つ補正アーキテクチャである。投資家にとって、この区別は、ランキング上の順位や、あるモデルのリリースに対する一時的な優位性ではなく、持続性を伴うデジタル条件を解消する価値に目を向けることを意味する。

AIの能力が高まるにつれ、その作業は、より多くの応用と運用判断に及ぶようになる。本稿で提示する商業的な考え方は、この拡大する環境において、独自の補正能力が価値ある役割を持つというものである。ARIS-ECは、より強力なAIと相補的な関係を持ち、同時に、非AI分野の応用が別の機会を提供する。

したがって、このアーキテクチャは、相互に関連する二つの水準で理解されるべきである。広い基礎研究体系としてのARと、焦点を定めた商用化の道筋としてのARIS-ECである。最初の応用は利用上の接点を示すが、基礎となる技術の境界を定義するものではない。

9.2 顧客価値は、繰り返す運用コストから始まる

再発するエラーは、最初の修復作業だけで終わらない。企業内の例を考えると、繰り返す調査と検討、データの再処理、作業の中断、システムに対する不確実性の再発を招き得る。維持条件の解消は、その循環が続く理由に対処する。

ARIS-ECが目指す事業価値は、繰り返す補正の削減、運用判断の根拠の明確化、正当なシステム動作の保持、そして異なる応用に統合できる補正プロセスである。これらは、アーキテクチャの目的を、顧客の立場から直接表したものである。

メトリクスは、この事業上の対話を支える。技術チームは、条件とその補正を確認する必要がある。運用チームは、推奨される措置と、それを誰が管理するかを把握する必要がある。事業責任者は、繰り返す作業への影響と、信頼できる運用を維持する費用を理解する必要がある。同じ追跡可能な結果が、これらの視点を結び付ける。

9.3 再利用と、保護された実装

APIとSDKによる道筋は、民間の応用分野で補正コアを再利用するための経路となる。顧客固有の統合に関連するシステムの文脈を提供し、基礎となるARIS-ECアーキテクチャが補正の規律を提供する。

Altheaの保護された枠組み、実装ノウハウ、管理された統合経路が、商用化戦略の基盤となる。公開資料は、アプローチとその実用的意義を説明する。適格な相手との技術的な検討では、コア実装の一般公開を必要とせず、承認された証拠と応用への適合性を調べられる。

9.4 投資が前進させるもの

資金は、文書化されたアーキテクチャを実行可能な製品にするための、ランタイムの実装、API・SDK統合、実証、管理された顧客評価を支える。これが開発の機会であり、より大きなモデルを求めて無期限に競争し続ける構想ではない。

投資の意義は、異なる技術的前提と、そこから構築される再利用可能な補正製品にある。

10. モデルの漸進的改善を越えるフロンティア

10.1 科学が扱う対象を変える

ARは、基礎的な転換を行う。持続性は、問題の中に観測される性質にとどまらず、それを維持する関係自体が探究の対象となる。起源、変換、再発、解消を一体として調べる。Transformative Mathematicsが形式的な研究体系を、ARISがその適用アーキテクチャを提供する。[1]

これによりARは、一つのソフトウェアカテゴリーを越える範囲を持つ。異なる科学・工学領域が一つの従来型メカニズムを共有していると主張することなく、それぞれに同じ持続性の問いを投げかけられる。ARIS-ECは、この広い方向性をデジタルシステムに絞って表したものである。

したがって、このフロンティアは、単なる高速な回答生成ではない。持続性を伴う問題とは何か、それを解消するとは何を意味するかを理解するための、異なる基盤である。この違いは、AIが進歩し続けても意味を保つ。

10.2 結論

AIとARIS-ECは、多くの同じ実用環境に貢献できるため、同じ議論の中で扱う意義がある。しかし、同じ技術として扱うべきではない。

現代のAIは、調査、創造、推論、発見の能力を拡大する。ARは、起源と、持続性を維持する構造から始まる。ARISは、そのアプローチをランタイムに実装し、ARIS-ECは、その中でデジタルシステム向けの制約付き補正・検証プロファイルを提供する。

科学者にとっての意義は、新たな探究対象と方法にある。エンジニアにとっては、条件、解消プロセス、測定値、結果を結び付ける補正アーキテクチャである。ビジネスと投資の関係者にとっては、特定のAIモデルを越える応用の広さを備えた、独自の製品化の方向である。

次のフロンティアは、持続性を伴うエラーにより効果的に対処することだけではない。その持続性の基盤を解消し、得られた補正を、理解可能で、追跡可能で、利用可能なものにすることである。

開発状況 — 2026年9月

ARとARISランタイムは、形式的検証およびエンジニアリング開発の段階にあり、実行可能なARIS-ECとAPI・SDKの提供を製品化の方向としている。本稿は、枠組み、文書化されたアーキテクチャ、応用上の役割を説明する。シナリオは例示であり、顧客への導入実績を報告するものではない。本番運用の性能と比較ベンチマークは、文書化された結果が得られた段階で公表する。

著者について

本稿の共同著者は、Christopher CurrellとSynthia Anariである。Christopher Currellは、ARの起源となる枠組み、科学的方向性、Althea Projectの指揮を担う。Synthia Anariは、アーキテクチャの統合的整理、技術的説明、原稿作成を担う。Althea Project GKのために共同で作成した。

用語集

Absolute Reductionism（絶対還元主義）／AR。 望ましくない持続性を、弁別・追跡・解消・検証の対象となる維持された構造として扱う、Althea Projectの枠組み。[1]

Transformative Mathematics（変容数学）。 持続性が生じ、解消される際の変換と維持関係を扱う、ARの形式的な数学研究体系。[1]

ARIS。 Absolute Reduction Integration Sequence。入力受け入れ、検出、分類、解消、検証、追跡可能な出力を結び付けるランタイムアーキテクチャ。[1]

ARIS-EC / Absolute Reduction Integration Sequence — Error Correction（絶対還元統合シーケンス — エラー補正）。 ARISランタイムのうち、デジタルシステムの補正と検証を行う制約付きプロファイル。最終的な補正の実行は顧客の管理下に置かれる。

持続性。 ある条件を維持する差異または関係が作用中、あるいは未解消であるために、その条件が継続して存在すること、または再発すること。有用な持続性は、望ましくない補正対象とは区別される。[1]

Postulate。 ARの公開用語では、条件を成立させる起源の仕様、または原因となる指示。[1]

Postulate Vector。 システムや状態の系列を通じて続く、postulateの方向性と影響。[1]

持続性を維持する構造。 対象の条件を維持する関係、指示、変換、境界、強化作用。[1]

解消。 対象とする望ましくない条件について、抑制、相殺、補正を繰り返す必要がなくなるように、その維持構造を終結させること。[1]

残差。 補正プロセスのある時点で、測定上残っている差異。

再発。 見かけ上の補正後に、条件が戻ること、または再生成されること。ARは、それを継続している持続性についての情報として扱う。[1]

検証。 関連する状態と再発の確認を含め、結果が、明示された範囲内で補正課題の要件を満たすと判断すること。

顧客が権限を付与したオペレーター。 顧客の導入方針に基づき、必要な証拠を検討し、該当する最終的な補正措置を承認する権限を割り当てられた人物。

API / SDK。 アプリケーション・プログラミング・インターフェース／ソフトウェア開発キット。顧客アプリケーションを補正ランタイムへ接続する統合インターフェースと、それを支える開発者向けコンポーネント。

参考文献と出典について

ARの概念的説明は、Altheaの公開科学論文[1]に基づく。ARIS-ECに関する記述は、Altheaの文書化されたアーキテクチャを著者らが公開向けに説明したものである。外部文献は、AIの事例と形式的ツールの説明を裏付けるものであり、引用先の組織によるAltheaへの支持を意味しない。公開資料の確認日は2026年9月24日である。

[1] Althea Project. *Persistence as a Resolvable Structure: The Scientific Proposition of Absolute Reductionism*. Official Public Scientific Paper、2026年7月改訂版。Althea Project公開文書。

[2] OpenAI. *Codex Security: now in research preview*. 2026年3月6日。

<https://openai.com/index/codex-security-now-in-research-preview/>

[3] Google. *A summer of security: empowering cyber defenders with AI*. 2025年7月15日。Big SleepによるSQLiteの事例を含む。

<https://blog.google/innovation-and-ai/technology/safety-security/cybersecurity-updates-summer-2025/>

[4] Novikov, A., Vű, N., Eisenberger, M., et al. *AlphaEvolve: A coding agent for scientific and algorithmic discovery*. arXiv:2506.13131、2025年6月16日。

<https://arxiv.org/abs/2506.13131>

[5] Google DeepMind. *AlphaEvolve: How our Gemini-powered coding agent is scaling impact across fields*. 2026年5月7日。

<https://deepmind.google/blog/alphaevolve-impact/>

[6] Feng, T., Trinh, T. H., Bingham, G., et al. *Towards Autonomous Mathematics Research*. arXiv:2602.10177。初回投稿：2026年2月10日。第3版：2026年3月6日。

<https://arxiv.org/abs/2602.10177>

[7] Lean. *The Lean Language Reference: Validating a Lean Proof*. 公式文書。閲覧日：2026年9月24日。

<https://lean-lang.org/doc/reference/latest/ValidatingProofs/>

[8] Dafny. *Dafny Reference Manual*. 公式文書。閲覧日：2026年9月24日。

<https://dafny.org/dafny/DafnyRef/DafnyRef>

発行情報

『人工知能とARIS-EC：問題解決、エラー補正、デジタルシステムの信頼性に対する異なるアプローチ』。Christopher Currell, Synthia Anari。Althea Project GK。第2.0版、2026年9月24日。日本語版。

本公開論文は、保護対象の方程式や機密の実装詳細を開示せず、枠組みと応用上の意義を説明する。