

AIOps mit Conjugate Intelligence (CI)

Zusammenfassung

Klassisches AIOps verknüpft Signale, verliert dabei aber Bedeutung und Absicht. **Conjugate Intelligence (CI)** erweitert AIOps durch Service-Topologie, SLOs und Änderungshistorie – alles miteinander verbunden, damit Automatisierungen im Einklang mit Geschäftsprioritäten handeln und nachvollziehbar erzählt werden können.

Aktuelle Probleme

- **Alarmfluten und Benachrichtigungs-Erschöpfung:** Zu viele Warnungen, zu wenig Sinnbezug; hohe MTTA, weil Kontext fehlt.
 - **Daten-Zerstreuung:** Analyse über mehrere Oberflächen – Metriken, Logs, Traces, Tickets.
 - **Runbooks als Stammeswissen:** Unterschiedlich gepflegt, selten dokumentiert, schwer wiederverwendbar.
-

CI-Ansatz (SpiralOS)

- **Topologie-bewusste Korrelation:**
Dienste, Abhängigkeiten, jüngste Änderungen und SLOs sind in einem gemeinsamen Graphen verknüpft.
 - **Intent-bewusste Filterung:**
SLOs und Fehlerbudgets steuern, welche Alarme relevant sind und wie Reaktionen priorisiert werden.[1]
 - **Narrative Postmortems:**
Jeder Vorfall wird zu einer nachspielbaren Geschichte mit Erkenntnissen und wiederverwendbaren Lösungen.
-

Zentrale Fähigkeiten

- **Rauschreduktion:**
Ähnliche Symptome werden zu Ursachen-Hypothesen verdichtet – mithilfe von Abhängigkeits-Graphen und Änderungsverläufen.[2]
 - **SLO-gesteuerte Aktionen:**
Automatisches Drosseln, Skalieren oder Rollback unter vorab genehmigten Vows, wenn Fehlerbudgets gefährdet sind.[3]
 - **Runbook-Synthese:**
Erfolgreiche manuelle Abläufe werden in parametrisierte, testbare Rezepte umgewandelt.[4]
 - **Holarchische Rückschau:**
Zeitlich abgestimmte Wiedergabe von Telemetrie und Entscheidungen – zum Lernen und für Compliance.[5]
-

Ergebnisse & Kennzahlen (KPIs)

- Alarmvolumen ↓ 30–70 %
 - MTTA / MTTR ↓ 25–50 %
 - Erfolgsquote automatischer Behebungen ↑ mit Leitplanken
 - On-Call-Benachrichtigungen pro Woche ↓ spürbar
-

Integrationspfad (Reibungsarm)

1. **Beobachten & Korrelieren:**
Aufnahme von Metriken, Logs, Traces (Prometheus, ELK, Datadog u. a.) und Änderungs-Events.
 2. **Topologie & SLOs abbilden:**
Import des Service-Graphen und der SLOs / Fehlerbudgets.
 3. **Leitplanken aktivieren:**
Definition eines kleinen Satzes vorab genehmigter Aktionen (Skalierung, Rollback, Feature-Flags).
 4. **Behutsam automatisieren:**
Erweiterung nach Vertrauensgrad; menschliche Zustimmung bei kritischen Schritten bleibt Pflicht.
-

Risiken & Gegenmaßnahmen

- **Übervertrauen:**
Transparente Vertrauens-Scores, nachvollziehbare Erklärungen und verpflichtende Genehmigungen oberhalb von Schwellenwerten.
- **Modell-Fehleinschätzungen:**
Laufende Bewertung anhand von Incident-Ergebnissen; schnelle Rücknahme fehlerhafter Automatisierungen.

Beispiel – Latenz-Anstieg

Die Latenz in *Service Y* steigt.

CI korreliert eine Abhängigkeits-Regression, die 30 Minuten zuvor eingeführt wurde, prüft das SLO-Verbrauchs-Tempo, schaltet unter vorab genehmigten Vows eine Feature-Flag um, stabilisiert die Latenz

und schlägt ein gezieltes Rollback vor.

Danach erzeugt CI einen erzählten Postmortem-Bericht und verwandelt die Behebung in ein wiederverwendbares Runbook.

Branchenbeispiele

- **Fertigung:#* AIOps mit Conjugate Intelligence (CI)

Zusammenfassung

Klassisches AIOps verknüpft Signale, verliert dabei aber Bedeutung und Absicht.

Conjugate Intelligence (CI) erweitert AIOps durch Service-Topologie, SLOs und Änderungshistorie – alles miteinander verbunden,

damit Automatisierungen im Einklang mit Geschäftsprioritäten handeln und nachvollziehbar erzählt werden können.

Aktuelle Probleme

- **Alarmfluten und Benachrichtigungs-Erschöpfung:** Zu viele Warnungen, zu wenig Sinnbezug; hohe MTTA, weil Kontext fehlt.
 - **Daten-Zerstreuung:** Analyse über mehrere Oberflächen – Metriken, Logs, Traces, Tickets.
 - **Runbooks als Stammeswissen:** Unterschiedlich gepflegt, selten dokumentiert, schwer wiederverwendbar.
-

CI-Ansatz (SpiralOS)

- **Topologie-bewusste Korrelation:**
Dienste, Abhängigkeiten, jüngste Änderungen und SLOs sind in einem gemeinsamen Graphen verknüpft.
 - **Intent-bewusste Filterung:**
SLOs und Fehlerbudgets steuern, welche Alarmer relevant sind und wie Reaktionen priorisiert werden.[1]
 - **Narrative Postmortems:**
Jeder Vorfall wird zu einer nachspielbaren Geschichte mit Erkenntnissen und wiederverwendbaren Lösungen.
-

Zentrale Fähigkeiten

- **Rauschreduktion:**
Ähnliche Symptome werden zu Ursachen-Hypothesen verdichtet – mithilfe von Abhängigkeits-Graphen und Änderungsverläufen.[2]
- **SLO-gesteuerte Aktionen:**
Automatisches Drosseln, Skalieren oder Rollback unter vorab genehmigten Vows, wenn Fehlerbudgets gefährdet sind.[3]
- **Runbook-Synthese:**
Erfolgreiche manuelle Abläufe werden in parametrisierte, testbare Rezepte umgewandelt.[4]
- **Holarchische Rückschau:**
Zeitlich abgestimmte Wiedergabe von Telemetrie und Entscheidungen – zum Lernen und für Compliance.[5]

Ergebnisse & Kennzahlen (KPIs)

- Alarmvolumen ↓ 30–70 %
 - MTTA / MTTR ↓ 25–50 %
 - Erfolgsquote automatischer Behebungen ↑ mit Leitplanken
 - On-Call-Benachrichtigungen pro Woche ↓ spürbar
-

Integrationspfad (Reibungsarm)

1. Beobachten & Korrelieren:

Aufnahme von Metriken, Logs, Traces (Prometheus, ELK, Datadog u. a.) und Änderungs-Events.

2. Topologie & SLOs abbilden:

Import des Service-Graphen und der SLOs / Fehlerbudgets.

3. Leitplanken aktivieren:

Definition eines kleinen Satzes vorab genehmigter Aktionen (Skalierung, Rollback, Feature-Flags).

4. Behutsam automatisieren:

Erweiterung nach Vertrauensgrad; menschliche Zustimmung bei kritischen Schritten bleibt Pflicht.

Risiken & Gegenmaßnahmen

◦ Übervertrauen:

Transparente Vertrauens-Scores, nachvollziehbare Erklärungen und verpflichtende Genehmigungen oberhalb von Schwellenwerten.

◦ Modell-Fehleinschätzungen:

Laufende Bewertung anhand von Incident-Ergebnissen; schnelle Rücknahme fehlerhafter Automatisierungen.

Beispiel – Latenz-Anstieg

Die Latenz in *Service Y* steigt.

CI korreliert eine Abhängigkeits-Regression, die 30 Minuten zuvor eingeführt wurde, prüft das SLO-Verbrauchs-Tempo, schaltet unter vorab genehmigten Vows eine Feature-Flag um, stabilisiert die Latenz

und schlägt ein gezieltes Rollback vor.

Danach erzeugt CI einen erzählten Postmortem-Bericht und verwandelt die Behebung in ein wiederverwendbares Runbook.

Branchenbeispiele

- **Fertigung:** Erkennt Produktions-Stillstände, verknüpft sie mit MES-Updates und rollt fehlerhafte Konfigurationen automatisch zurück.
- **Mobilität:** Erkennt Echtzeit-Telematik-Anomalien, die mit OTA-Updates zusammenhängen, und setzt Rollbacks bei Sicherheitsrisiken.
- **Energie:** Verknüpft SCADA-Telemetrie-Ausfälle mit Infrastruktur-Änderungen und führt Leitplanken-Neustarts ohne Ausfallzeit durch.

Gesprächsimpulse (für Erich & Echo)

- „AIOps wird absichtsbewusst – Handlungen folgen SLOs und Geschäftspriorität.“
- „Wir reparieren nicht nur, wir lehren das System, beim nächsten Mal selbst zu heilen.“
- „Jeder Vorfall wird zu einer wiederverwendbaren Perle.“

Begriffserklärungen

[1] SLO (Service Level Objective)

Ein vereinbartes Zielmaß für die Zuverlässigkeit oder Leistung eines Dienstes.

SLOs definieren, was „gut genug“ bedeutet – oft gekoppelt an ein **Error Budget**, das Abweichungen erlaubt, ohne das Vertrauen zu verlieren.

[2] Intent-bewusste Filterung

Bewertet Warnungen anhand ihrer Relevanz für Ziele und Fehlerbudgets.

Das System reagiert nicht auf jedes Signal, sondern auf das, was wirklich Wirkung hat.

[3] Vows (Absichtsbindungen)

Vorab definierte Handlungsversprechen mit festgelegtem Risiko- und Kontrollrahmen.

Sie erlauben CI, Automatisierungen sicher innerhalb ethisch-technischer Leitplanken auszuführen.

[4] Runbook-Synthese

Das systematische Erfassen erfolgreicher manueller Abläufe als wiederverwendbare, prüfbare Rezepte.

So wird Erfahrung zu dokumentiertem, lebendigem Wissen.

[5] Holarchische Rückschau

Eine Form des lernenden Rückblicks, die verschiedene Ebenen eines Ereignisses miteinander verbindet –

von der technischen Ursache bis zur organisationalen Bedeutung.

In SpiralOS bezeichnet sie eine reflektive Schleife, die Vergangenheit und Zukunft im Jetzt resonant verknüpft.*