### Documento Programmatico: Gestione Event-Driven per l'Evoluzione della Tassonomia MIU

**Data:** 11 Luglio 2025

**1. Visione Fondamentale: Emulare la Fluidità Biologica dell'Apprendimento**

La visione alla base dell'evoluzione della tassonomia nel sistema MIU si ispira al comportamento del cervello biologico, in particolare al meccanismo sinaptico. L'obiettivo è superare la logica di aggiornamento discreta e temporizzata (es. "ogni X ore" o "su richiesta manuale") per adottare un approccio più dinamico e intrinseco.

Nel cervello, l'apprendimento e la riorganizzazione delle informazioni non avvengono in cicli rigidi, ma come un flusso continuo di eventi. L'apertura o la chiusura di una sinapsi è un fatto fisico deterministico, ma la **significazione** che emerge da una catena di questi eventi è fluida e complessa, simile all'acqua che si espande nel mare. Il nostro sistema MIU deve emulare questa capacità di auto-organizzazione e adattamento continuo, reagendo a stimoli interni derivanti dalla propria esperienza.

**2. Il Concetto di "Evento" come "Fatto Fisico" Discreto**

Nel contesto del sistema MIU, un "evento" non è la "significazione" complessa, ma la **registrazione di un singolo "fatto fisico" discreto** avvenuto in un preciso istante. Questi eventi sono gli "impulsi nervosi" o i "neurotrasmettitori" del nostro sistema:

* **RuleAppliedEvent**: Segnala che una regola MIU è stata applicata, includendo dettagli sull'applicazione (ID ricerca, stati coinvolti, regola applicata, profondità, esito immediato).
* **SearchCompletedEvent**: Indica la conclusione di un'intera ricerca MIU, con il suo esito finale (successo, fallimento per limiti, fallimento per stallo, ecc.).
* **AnomalyDetectedEvent**: Segnala la rilevazione di un comportamento inatteso o inefficiente da parte del sistema di rilevamento anomalie.

Queste classi evento sono semplici contenitori di dati, il "mezzo" per trasportare l'informazione grezza.

**3. La "Significazione" Emergente dal Flusso di Eventi e dall'Interpretazione**

La vera "significazione" e la "fluidità biologica" emergono non dal singolo evento statico, ma dalla **dinamica e dal contesto del flusso continuo di questi eventi**. È l'interpretazione di sequenze, frequenze e combinazioni di eventi che permette al sistema di "comprendere" il proprio stato di apprendimento.

Questo processo di interpretazione è gestito da un componente chiave: il **TaxonomyOrchestrator**.

**4. Il Ruolo del TaxonomyOrchestrator (La Rete di Petri): Quando Scatta l'Evoluzione**

Il TaxonomyOrchestrator incarna la logica di una **rete di Petri**, fungendo da "sistema nervoso" centrale che monitora l'attività di apprendimento di MIU.

* **Raccolta di "Token":** Ogni volta che un "fatto fisico" (un evento) viene pubblicato nel sistema, il TaxonomyOrchestrator lo traduce in un **"token"** che viene depositato in un "luogo" specifico all'interno della sua rete di Petri interna. Questi luoghi possono rappresentare:
  + Nuove applicazioni di regole.
  + Ricerche completate (distinguendo successi da diversi tipi di fallimenti).
  + Anomalie rilevate.
  + Il tempo trascorso.
* **Gestione di Coincidenze e Sincronie:** Le "transizioni" all'interno della rete di Petri del TaxonomyOrchestrator sono configurate per "scattare" (attivarsi) solo quando specifiche **combinazioni e quantità di token** sono presenti nei loro luoghi di input. Questo permette al sistema di riconoscere pattern complessi di attività e di "sentire" quando è il momento opportuno per una riorganizzazione.
* **Il "Collasso della Funzione d'Onda":** Quando una transizione nel TaxonomyOrchestrator scatta, essa rappresenta il "collasso della funzione d'onda" della conoscenza potenziale. Questo è il **momento deterministico** in cui la conoscenza potenziale si cristallizza.
* **Innesco del RuleTaxonomyGenerator:** Lo scatto di una transizione significativa nel TaxonomyOrchestrator ha come effetto l'invocazione del RuleTaxonomyGenerator. Questo è il punto in cui la tassonomia viene rigenerata o aggiornata, trasformando l'informazione fluida in una struttura organizzata e utilizzabile.

**5. La Gestione del Fallimento: Informazione, non Solo Assenza**

Un aspetto cruciale di questo approccio è la gestione del fallimento. A differenza di un successo (che è un evento binario), il fallimento è una condizione di incertezza e può avvenire in molteplici modi (es. limiti raggiunti, stallo, timeout, nessuna regola applicabile).

* I log dettagliati del sistema sono essenziali per capire la natura del fallimento.
* Queste informazioni vengono incapsulate in **eventi differenziati** (es. SearchCompletedEvent con diversi Outcome o AnomalyDetectedEvent con vari AnomalyType).
* Il TaxonomyOrchestrator utilizza questi token specifici per interpretare la "significazione" del fallimento, permettendo alla tassonomia di apprendere non solo dai successi, ma anche dalle sfumature delle difficoltà incontrate.

**6. Architettura Event-Driven e Parallelismo**

Questo modello promuove un'architettura **fortemente disaccoppiata e parallela**:

* I componenti (es. MIUDerivationEngine, MIUExplorer, AnomalyDetectionManager) pubblicano eventi senza sapere chi li ascolterà.
* I sottoscrittori (es. un servizio di persistenza che scrive nel database, il TaxonomyOrchestrator che genera token) reagiscono agli eventi in modo indipendente e potenzialmente concorrente.

Questo garantisce che l'aggiornamento della tassonomia e la persistenza dei dati possano avvenire in parallelo, migliorando la reattività e la "fluidità" complessiva del sistema, avvicinandolo al comportamento di un sistema biologico.