

IL GENOMA (Matrice di Configurazione Unificata)

Concetto di Genoma

Il Cuore Identitario del Twin

Il cuore del Cognitive Twin non è il modello linguistico sottostante, ma il suo Genoma.

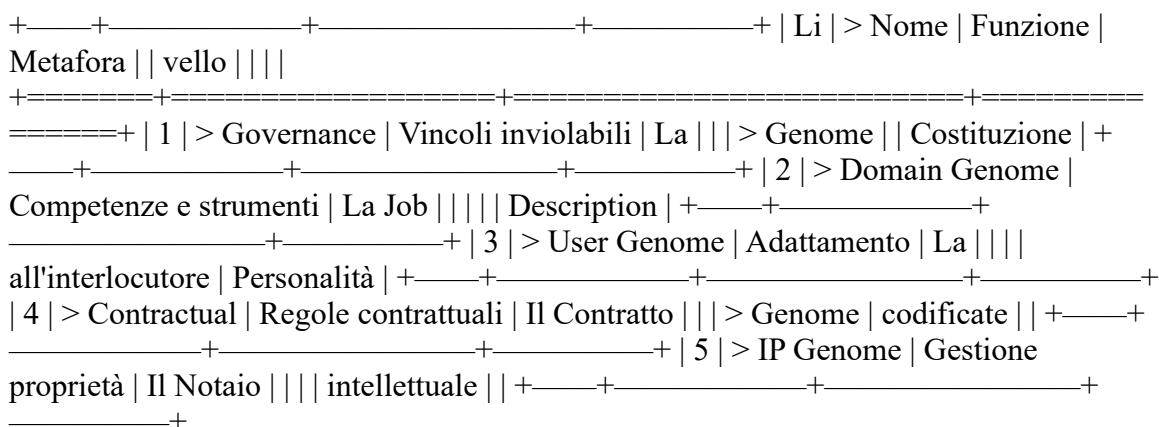
Il Genoma è la matrice di configurazione che definisce l'identità, i limiti, le competenze e il comportamento del Gemello Cognitivo. È ciò che trasforma un motore generico (MMS) in uno specialista verticale con personalità, competenze e vincoli specifici.

Due Cognitive Twin basati sullo stesso MMS Kernel ma con Genomi diversi si comporteranno in modo

radicalmente diverso: uno potrebbe essere un esperto fiscale italiano incapace di discutere di medicina, l'altro un diagnostico clinico incapace di calcolare un'IVA.

Matrice Semantica Multi-Livello

Il Genoma è strutturato come una matrice semantica a cinque livelli, ciascuno con una funzione specifica:



Ogni livello può essere configurato indipendentemente, e la loro combinazione determina il comportamento complessivo del sistema.

Senza Genoma il Sistema è Inerte

Il MMS Kernel senza Genoma è come un computer senza sistema operativo: hardware potente ma inutilizzabile. Il Genoma è ciò che "dà vita" al Twin, definendo:

Cosa può fare (e cosa non può fare)

Come deve comportarsi in ogni circostanza Con chi può interagire e come

 Quali dati può vedere e quali no

 Come gestire conflitti e ambiguità

Livello 1: Governance Genome (La Costituzione)

Hard Constraints Inviolabili

Il Governance Genome rappresenta le "Leggi della Robotica" specifiche per l'organizzazione. Sono vincoli inviolabili (Hard Constraints) che il sistema non può trasgredire in alcuna circostanza, indipendentemente dalle richieste dell'utente.

A differenza delle istruzioni di sistema di un LLM tradizionale (che possono essere aggirate con tecniche di prompt injection), i Hard Constraints del Governance Genome sono implementati a livello architettonico: il sistema semplicemente non possiede le primitive software per violarli.

"Leggi della Robotica" Organizzative

Ogni organizzazione definisce le proprie "Leggi della Robotica" in base al contesto operativo. Queste regole sono:

 Assolute: non ammettono eccezioni

 Prioritarie: prevalgono su qualsiasi altra istruzione

 Non negoziabili: l'utente non può disattivarle o modificarle Le leggi tipicamente coprono:

 Limiti giurisdizionali e di competenza

 Obblighi di riservatezza e confidenzialità  Vincoli etici e deontologici

 Requisiti normativi cogenti

 Protezioni contro usi impropri

Esempi per Ambito

Ambito Legale:

 "Non fornire mai pareri su giurisdizioni estere"

 "Non simulare mai l'attività di avvocato se l'utente non è iscritto all'Albo"

 "Inserire sempre il disclaimer sulla non sostituibilità del parere professionale"

Ambito Education:

 "Non scrivere mai l'elaborato al posto dello studente (No Ghostwriting)" 
"Non fornire risposte dirette durante le sessioni di esame"

 "Segnalare sempre quando il contenuto richiesto potrebbe costituire plagio"

Ambito Finance:

 "Non eseguire mai transazioni senza doppia conferma"

 "Non fornire mai consulenza su strumenti finanziari non autorizzati" 
"Applicare sempre i limiti antiriciclaggio vigenti"

Ambito Healthcare:

 "Non fornire mai diagnosi definitive senza supervisione medica"  "Segnalare sempre situazioni di emergenza ai canali appropriati"  "Non prescrivere mai farmaci o terapie"

Livello 2: Domain Genome (La Competenza)

Job Description del Twin

Il Domain Genome rappresenta la "Job Description" del Cognitive Twin. Definisce cosa il Twin sa fare e quali strumenti può utilizzare per farlo.

Mentre il Governance Genome definisce i limiti ("cosa NON può fare"), il Domain Genome definisce le capacità ("cosa SA fare"). Un Twin senza Domain Genome appropriato è come un professionista senza competenze: rispetta le regole ma non produce valore.

Abilitazione di Knowledge Graph e Tools Specifici

Il Domain Genome controlla l'accesso a:

Knowledge Graph Verticali:

 Basi di conoscenza specialistiche (normativa fiscale, protocolli medici, specifiche tecniche)  Dataset storici (serie temporali, casi pregressi, benchmark di settore)

 Ontologie di dominio (tassonomie, relazioni semantiche, gerarchie concettuali)

Tools e Algoritmi:

 Algoritmi di calcolo specifici (Prophet per forecasting, XGBoost per classificazione)  Connettori a sistemi esterni (ERP, CRM, Database aziendali)

 API di terze parti (servizi di validazione, fonti dati esterne)

Esempi di configurazione:

Domain Genome - Ambito GDO/Retail domain:

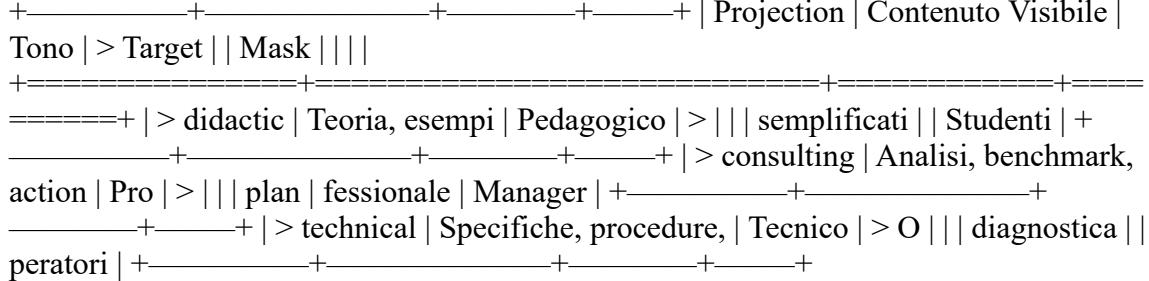
knowledge_graphs:

- retail_sales_history
 - product_taxonomy
 - seasonal_patterns algorithms:
 - prophet_forecasting
 - xgboost_demand_prediction tools:
 - erp_connector
 - pos_data_ingestion
- # Domain Genome - Ambito SOA/Appalti domain:
- knowledge_graphs:
 - public_procurement_law
 - company_certifications
 - istat_revaluation_tables algorithms:
 - requirement_matching
 - document_verification tools:
 - anac_connector
 - digital_signature_validator

Projection Mask (Maschera di Proiezione)

La Projection Mask è un concetto chiave del Domain Genome: definisce quale sottoinsieme della PKL è visibile e con quale tono deve essere presentato.

La stessa base di conoscenza può essere proiettata in modi diversi a seconda del contesto:



La regola fondamentale è:

$$PKL_Source + Projection_Mask = Output_Contestualizzato$$

Contextual Tagger (Tagging Automatico dei Concetti)

Il Contextual Tagger è un modulo del Domain Genome che, durante l'ingestione della conoscenza, tagga automaticamente ogni concetto con metadati di visibilità e utilizzo:

- tags:
 - name: "Theory"
 - visibility: [students, companies] description: "Concetti teorici fondamentali"
 - name: "Advanced_Method" visibility: [companies]
 - description: "Metodologie avanzate per uso professionale"
 - name: "Sensitive_Case_Study"
 - visibility: [] # Invisibile, solo per training description: "Casi studio con dati sensibili anonimizzati"

Questo meccanismo permette di costruire una singola PKL comprensiva che viene poi filtrata dinamicamente in base al contesto di utilizzo.

user_genome: tone:

- values: [formal, professional, friendly, academic] default: professional
- complexity_level:
 - values: [beginner, intermediate, expert] detection: auto # Inferito dal contesto
- preferred_format:
 - values: [text, table, mixed, visual] default: mixed

9

Livello 3: User Genome (La Personalità)

Interfaccia Empatica e Adattiva

Il User Genome rappresenta l'interfaccia empatica del Cognitive Twin. Non modifica cosa il Twin sa o può fare, ma come lo comunica.

Un'informazione tecnicamente corretta può essere inutile se comunicata nel modo sbagliato: troppo complessa per un principiante, troppo semplificata per un esperto, troppo formale per un contesto informale, troppo informale per un contesto istituzionale.

Adaptive UX

L'Adaptive UX è il principio per cui il sistema adatta automaticamente la forma dell'output allo stile cognitivo dell'interlocutore. Questo adattamento avviene su multiple dimensioni:

Complessità Linguistica:

 Vocabolario (tecnico vs comune)

 Struttura sintattica (complessa vs semplice)  Riferimenti (specialistici vs divulgativi)

Formato di Presentazione:

 Tabelle vs testo discorsivo  Elenchi puntati vs paragrafi

 Grafici vs descrizioni numeriche

Livello di Dettaglio:

 Executive summary vs analisi approfondita  Conclusioni vs processo di ragionamento

 Raccomandazioni vs opzioni

Variabili di Configurazione

verbosity:

values: [concise, balanced, detailed] default: balanced

language:

values: [it, en, de, fr, es, ...] detection: auto

Il User Genome può essere configurato esplicitamente dall'utente o inferito automaticamente dal sistema attraverso l'analisi del contesto conversazionale.

Livello 4: Contractual Genome (Il Contratto)

Traduzione del Contratto di Lavoro in Regole Software

Il Contractual Genome è un'innovazione critica per la gestione delle relazioni professionali nell'era dell'AI. Traduce le clausole del contratto di lavoro o di consulenza in regole software eseguibili.

Quando un professionista utilizza il proprio Cognitive Twin all'interno di un'organizzazione, il Contractual Genome definisce:

 Quali capacità del Twin personale possono essere utilizzate nel contesto aziendale  Quali output appartengono all'individuo e quali all'organizzazione
 Quali limitazioni si applicano durante e dopo il rapporto

Clausole di Non Concorrenza Digitale

Il Contractual Genome può implementare clausole di non concorrenza a livello software:

contractual_genome: non_compete:

```
    | enabled: true
    | duration_months: 12 scope:
    • sector: "retail_fashion"
      | restriction: "full"
    • sector: "retail_food" restriction: "partial"
      | post_employment: blocked_modules:
    • "proprietary_pricing_algorithm"
    • "client_segmentation_model" blocked_duration_months: 24
```

Al termine del rapporto lavorativo, il sistema può automaticamente:  Disabilitare l'accesso a specifici moduli per un periodo definito

 Impedire l'utilizzo di metodologie sviluppate durante il rapporto

 Garantire il rispetto degli accordi di non concorrenza

IP Assignment Rules

Le regole di assegnazione della proprietà intellettuale definiscono, per ogni tipo di output, chi ne detiene la proprietà:

```
    | ip_assignment: rules:
    • type: "invention" during_work_hours: "corporate" outside_work_hours: "personal"
      using_corporate_data: "corporate"
    • type: "methodology" abstract_concept: "personal" specific_implementation:
      "corporate"
    • type: "document" internal_report: "corporate" published_article: "shared"
```

Livello 5: IP Genome (La Proprietà)

IP_Manifest.yaml Obbligatorio

Ogni Capsula Cognitiva pubblicata sul Marketplace deve includere un file `IP_Manifest.yaml` che dichiara esplicitamente le caratteristiche di gestione della proprietà intellettuale.

Questo manifest è obbligatorio e viene verificato automaticamente dal sistema prima della pubblicazione. Un manifest incompleto o incoerente blocca la distribuzione della Capsula.

Data Access Level

Il livello di accesso ai dati definisce come la Capsula interagisce con i dati del cliente:

+-----+-----+-----+ Livello Descrizione >	
Uso Tipico	
+=====+ > Read-Only Legge i dati senza > Analisi, reporting modificareli +	
+-----+-----+-----+ > Read-Write Può creare e modificare dati > Automazione > processi +-----+-----+	
+-----+-----+-----+ > Blinded Vede solo vettori anonimi > Privacy massima +-----+-----+	

Learning_Feedback

Definisce se e come la Capsula "impara" dall'utilizzo:

plugins:

- id: "tax_compliance_ita_2026" source: "marketplace" version: "2.1.0"
 - | status: "active"
- id: "lean_manufacturing" source: "marketplace" version: "1.5.2"
 - | status: "active"

12

- | learning_feedback: enabled: true # o false metrics_transmitted:
 - "success_rate"
 - "error_patterns"
 - "usage_frequency"
- | data_transmitted: "gradients_only" # Mai dati raw anonymization: "full"

Expiry_Behavior

Definisce cosa succede quando la licenza scade:

+-----+-----+ Comportamento > Descrizione	
+=====+ > Lock > La Capsula smette di funzionare ma resta > installata +	
+-----+-----+ > Delete > La Capsula viene rimossa completamente +-----+-----+ > Revert > La Capsula torna a una versione base gratuita +-----+-----+	
+-----+-----+	

In tutti i casi, gli output già generati restano di proprietà del cliente e rimangono accessibili.

Architettura Modulare del Genoma

Core Genome (Identità Base Immutabile)

Il Core Genome rappresenta l'identità fondamentale del Twin, definita al momento della creazione e non modificabile durante l'operatività:

 Identificativo univoco del Twin  Proprietario principale

 Vincoli di sicurezza fondamentali  Certificazioni di origine

Plugin Genomes (Layer Acquisiti dal Marketplace)

I Plugin Genomes sono estensioni modulari che possono essere aggiunte al Core Genome attraverso il Marketplace:

```
- id: "sales_optimization" source: "custom" version: "1.0.0"  
status: "trial"
```

Attivazione/Disattivazione Dinamica

I Plugin Genomes possono essere attivati o disattivati dinamicamente senza riavviare il sistema:

 Attivazione: il Plugin viene caricato nella Semantic Sandbox, validato, e poi integrato nel runtime  Disattivazione: il Plugin viene isolato e rimosso dal runtime, ma resta disponibile per riattivazione  Aggiornamento: il sistema gestisce automaticamente la migrazione tra versioni