Article

Not peer-reviewed version

Modelling Intelligent Systems Using COH: A Unified Framework for Intelligence and Intelligent Systems

Harris Wang *

Posted Date: 29 September 2025

doi: 10.20944/preprints202509.2371.v1

Keywords: artificial general intelligence; intelligent systems modelling; constrained object hierarchies; system design framework; adaptive systems; neuroscience-inspired AI

Article

Modelling Intelligent Systems Using COH: A Unified Framework for Intelligence and Intelligent Systems

Harris Wang

Athabasca University, Canada; harrisw@athabascau.ca

Abstract

Constrained Object Hierarchies (COH) present a neuroscience-grounded theoretical framework for artificial general intelligence (AGI) that enables unified modelling of intelligent systems across diverse application domains. This paper introduces the COH framework as a comprehensive 9-tuple formalism that captures the essential components of intelligent systems through compositional hierarchy, adaptive neural components, and constraint-based reasoning. We demonstrate the framework's versatility by modelling 15 distinct intelligent systems across domains including autonomous vehicles, healthcare, smart grids, personalized education, and cybersecurity. Each model illustrates how COH's innovative concepts—particularly identity constraints, goal constraints, trigger constraints, and constraint daemons—provide a common language for describing intelligent behavior. The paper establishes COH as a foundational theory for AGI system design, offering methodological guidance for analysis, design, and implementation of intelligent systems with provable properties and predictable behavior.

Keywords: artificial general intelligence; intelligent systems modelling; constrained object hierarchies; system design framework; adaptive systems; neuroscience-inspired AI

---

1. Introduction

Intelligent systems design has traditionally followed domain-specific paradigms, resulting in fragmented methodologies that hinder cross-domain knowledge transfer and unified theoretical development. The absence of a comprehensive framework for modelling intelligence has impeded progress toward Artificial General Intelligence (AGI), as current approaches often excel in narrow domains while lacking the flexibility and adaptability characteristic of natural intelligence. Constrained Object Hierarchies (COH) emerge as a neuroscience-grounded theoretical framework that addresses this gap by providing a unified formalism for modelling intelligent systems across diverse application domains.

The fundamental premise of COH is that intelligence arises from the interaction of hierarchical compositional structures with multiple constraint types that govern system behavior. This paper makes three primary contributions: (1) it formalizes the COH framework as a comprehensive 9-tuple model; (2) it demonstrates the framework's applicability through detailed modelling of 15 intelligent systems across distinct domains; and (3) it establishes methodological guidelines for implementing COH models as executable systems. Our work shows that COH provides not merely a modelling language but a foundational theory for understanding and engineering intelligent systems with predictable properties.

2. Literature Review

Research in intelligent systems modelling has evolved through several paradigms, from symbolic AI [1] to connectionist approaches [2] and hybrid systems [3]. Early work in hierarchical systems [4] and constraint satisfaction [5] laid groundwork for structured reasoning, while more recent advances in deep learning [6] and reinforcement learning [7] have demonstrated the power of

doi:10.20944/preprints202509.2371.v1

adaptive components. However, these approaches have typically addressed aspects of intelligence in isolation without providing a unified framework.

Neuroscience has increasingly informed AI research, with hierarchical cortical models [8] and predictive coding theories [9] offering biological plausibility for artificial intelligence architectures. The integration of compositional structure with neural adaptation represents a promising direction [10], yet existing frameworks lack the comprehensive constraint mechanisms necessary for modelling complex intelligent behavior. COH addresses this limitation by synthesizing hierarchical composition, neural adaptation, and multi-constraint governance into a coherent framework.

3. Constrained Object Hierarchies (COH) Framework

3.1. Theoretical Foundations

Constrained Object Hierarchies present a neuroscience-grounded theoretical framework that conceptualizes intelligent systems as hierarchical compositions of objects governed by multiple constraint types. The framework is grounded in several key principles derived from cognitive science and neuroscience: the hierarchical organization of cerebral cortex [8], the predictive processing theory of brain function [9], and the constraint-satisfaction nature of neural computation [11]. COH integrates these principles into a unified formalism that captures both the structural and functional aspects of intelligent systems.

The COH framework addresses a fundamental challenge in AGI research: how to create systems that exhibit flexibility, adaptability, and general problem-solving capabilities while maintaining stability and predictable behavior. By modelling intelligence as emerging from the interaction between hierarchical structure and constraint governance, COH provides a pathway to systems that can operate effectively across diverse domains and problem types. The framework's grounding in neural principles ensures biological plausibility while its formal rigor enables engineering implementation.

3.2. Formal Representation

A COH is formally represented as a 9-tuple: O = (C, A, M, N, E, I, T, G, D), where each component captures essential aspects of intelligent systems:

- C (Components): The compositional hierarchy of sub-objects that constitute the system. This reflects the fundamental principle that complex systems are built from simpler components organized in nested structures [12]. The hierarchy enables modular design and supports emergent behavior through component interactions.
- A (Attributes): The state variables that represent the system's current condition and memory. Attributes provide the informational substrate upon which the system operates and include both internal states and representations of external conditions.
- M (Methods): The executable actions that enable the system to interact with its environment and modify its state. Methods implement the system's operational capabilities and represent the behavioral repertoire available to achieve goals.
- N (Neural Components): Adaptive models that provide learning and pattern recognition capabilities. These components enable the system to improve its performance through experience and to handle uncertain or noisy information [6].
- E (Embedding): A neural component that creates semantic embeddings of the COH's overall state. This provides context awareness and enables the system to reason about its situation holistically, similar to cognitive maps in biological intelligence [13].

- I (Identity Constraints): Fundamental rules that define the system's essential character and inviolable principles. These constraints represent non-negotiable conditions that must always be satisfied, analogous to homeostatic mechanisms in biological systems [14].
- T (Trigger Constraints): Event-condition-action rules that govern reactive behavior. These constraints implement the system's ability to respond to specific situations with appropriate actions, providing stimulus-response capabilities while maintaining coordination with higher-level goals.
- G (Goal Constraints): Optimization objectives that guide system behavior toward desired outcomes. These constraints represent the system's motivational drivers and enable purposeful action through multi-objective optimization [15].
- D (Constraint Daemons): Real-time monitors that continuously evaluate constraint satisfaction and initiate corrective actions. Daemons provide meta-cognitive capabilities for self-monitoring and adaptive control, ensuring robust operation in dynamic environments.

4. Methodologies for COH-Based Modelling

4.1. General Principles

Modelling intelligent systems with COH follows several fundamental principles that ensure coherent and effective system design. The first principle is hierarchical decomposition, where complex systems are recursively broken down into simpler components until reaching primitive elements. This approach mirrors natural intelligent systems, which exhibit nested organizational structures from neural circuits to brain regions to entire cognitive systems [16]. The second principle is constraint prioritization, where identity constraints take precedence over all other constraint types, followed by trigger constraints, with goal constraints providing optimization guidance rather than hard requirements.

The methodology emphasizes the separation of concerns between the structural aspects (C, A, M), adaptive capabilities (N, E), and governance mechanisms (I, T, G, D) of intelligent systems. This separation enables modular design, where different aspects of the system can be developed, tested, and modified independently while maintaining overall coherence. The approach also supports incremental development, where systems can evolve from simple implementations to increasingly sophisticated versions by enhancing individual components.

4.2. Analysis and Design Process

The COH-based design process begins with requirement analysis to identify the system's purpose, operating environment, and performance expectations. This analysis directly informs the specification of constraints: identity constraints derived from safety and integrity requirements, trigger constraints from operational scenarios, and goal constraints from performance objectives. The component hierarchy is then designed through functional decomposition, identifying natural partitions of responsibility and information hiding boundaries.

Neural components are specified based on the types of uncertainty the system must handle, and the learning capabilities required. The embedding component is designed to capture the semantic relationships most relevant to the system's decision-making context. Constraint daemons are identified for critical monitoring functions, particularly those involving complex or evolving conditions that require continuous assessment. Throughout the design process, consistency checks ensure that constraints are mutually compatible and that the component hierarchy properly encapsulates the system's functional requirements.

5. Workflow for Domain-Specific Modelling

 doi:10.20944/preprints202509.2371.v1

5.1. Domain Analysis Phase

The initial phase of COH modelling involves comprehensive analysis of the application domain to identify characteristic challenges, performance requirements, and constraints. Domain analysis begins with identifying the key entities, relationships, and processes that define the domain's structure and dynamics. This analysis reveals the natural hierarchies that exist within the domain, which form the basis for the component structure (C) of COH models. For each domain, we identify typical sources of uncertainty and variability that necessitate adaptive capabilities, informing the design of neural components (N) and embedding mechanisms (E).

Critical to this phase is the identification of domain-specific constraints that govern system behavior. Identity constraints (I) are derived from fundamental laws, safety requirements, or ethical principles that must never be violated. Trigger constraints (T) are identified from common event-response patterns within the domain. Goal constraints (G) are extracted from performance objectives and optimization criteria relevant to the domain. This constraint analysis ensures that the resulting COH models capture the essential governance mechanisms that define intelligent behavior within the specific application context.

5.2. Model Construction Phase

Following domain analysis, the model construction phase translates domain knowledge into formal COH specifications. This phase employs iterative refinement, beginning with high-level component definitions and progressively elaborating detailed specifications. The construction process follows a pattern of defining components, their attributes and methods, then specifying the neural components needed for adaptation, and finally formalizing the constraint system that governs component interactions.

A key aspect of model construction is ensuring consistency across the constraint hierarchy. Identity constraints must be satisfiable by the component architecture, trigger constraints must align with available methods and detectable events, and goal constraints must be measurable through attribute values. The embedding component is designed to integrate information from across the hierarchy into a coherent situational representation that supports constraint evaluation and decision-making. The completed model provides a comprehensive specification that can guide implementation while maintaining the theoretical coherence of the COH framework.

6. Modelling of Intelligent Systems Across Domains

6.1. Autonomous Vehicles: Ego-Vehicle Perception & Control System

System Description and Significance: Autonomous vehicles represent a critical application domain where intelligent systems must operate safely in complex, dynamic environments. The ego-vehicle system integrates perception, decision-making, and control to navigate while avoiding hazards and complying with traffic regulations. This system's significance lies in its potential to transform transportation through improved safety and efficiency while presenting substantial technical challenges in real-time intelligence.

Analysis and Design: The system requires hierarchical decomposition into functional modules (perception, localization, planning, control) with precise coordination. Safety emerges as the paramount concern, dictating identity constraints that prevent collisions. The design must accommodate uncertain sensor data through neural components while maintaining predictable behavior through constraint governance. The embedding component provides situational awareness by integrating information across modules.

Modelling Solution:

C: {PerceptionModule, LocalizationModule, PlanningModule, ControlModule}

A: current_pose, velocity, sensor_data, trajectory_plan, obstacle_list

M: filter_sensor_data(), update_belief_state(), calculate_trajectory(), adjust_steering()

N: NeuralObjectDetector, NeuralMotionPredictor
E: Situation embedding unifying module outputs for holistic scene understanding
I: safety_bubble (minimum safe distance maintained at all times)
T: IF obstacle_detected WITHIN trajectory THEN invoke emergency_braking AND replan_path
G: Minimize travel_time AND Maximize passenger_comfort AND Adhere_to_traffic_laws
D: CollisionAvoidanceDaemon monitoring obstacle_list and trajectory

The COH model captures the essential intelligence requirements: hierarchical decomposition for modularity, neural components for handling perception uncertainty, identity constraints for safety, and goal constraints for multi-objective optimization. The constraint daemon provides continuous monitoring for proactive hazard avoidance.

6.2. Healthcare: Patient-Specific Treatment Recommender

System Description and Significance: This system personalizes medical treatment by integrating patient data, clinical knowledge, and outcome predictions. Its significance lies in enabling precision medicine, where treatments are tailored to individual patient characteristics rather than population averages. The system addresses the critical challenge of balancing treatment efficacy with safety considerations in complex medical decision-making.

Analysis and Design: The system must integrate diverse information sources while maintaining rigorous safety standards. Identity constraints encode fundamental medical ethics and safety protocols, while neural components enable personalization by learning from patient data and treatment outcomes. The hierarchical structure separates concerns between data management, knowledge representation, and recommendation logic.

Modelling Solution:
C: {PatientMedicalRecord, ClinicalGuidelines, DrugInteractionDatabase, RecommenderEngine}
A: patient_vitals, genomic_data, current_medications, treatment_options
M: query_guidelines(), check_interactions(), calculate_personalized_dose()
N: OutcomePredictionModel for treatment efficacy/side-effects
E: Embedding of patient's overall health status and disease profile
I: Recommended treatment must not have severe known interactions (safety_first)
T: IF new_lab_result indicates_toxicity THEN alert_doctor AND suggest_alternative
G: Maximize_treatment_efficacy AND Minimize_side_effects
D: SafetyDaemon cross-referencing new patient data with identity constraints

The COH formalism captures the delicate balance between personalized adaptation and safety assurance through its constraint hierarchy. The neural components enable precision medicine while the identity constraints maintain safety boundaries.

6.3. Smart Grids: Dynamic Load Balancer

System Description and Significance: Modern power grids require intelligent balancing of electricity supply and demand amid fluctuating renewable generation and consumption patterns. This system's significance lies in enabling efficient integration of renewable energy sources while maintaining grid stability—a critical requirement for sustainable energy infrastructure.

Analysis and Design: The system must coordinate multiple generation and distribution components while responding to real-time changes in supply and demand. Identity constraints encode physical laws of grid stability, while neural components predict load patterns and renewable generation. The hierarchical structure mirrors the physical organization of power grids from generation to consumption.

Modelling Solution:
C: {PowerGenerationNodes, DistributionSubstations, ConsumerZones}
A: power_generated, power_demanded, grid_frequency, line_capacity
M: reroute_power(), shed_non_critical_load(), adjust_generator_output()
N: LoadForecastModel, RenewableGenerationPredictor

 doi:10.20944/preprints202509.2371.v1

E: Embedding of overall grid stability and stress level
I: grid_frequency must remain within strict tolerance (physical stability)
T: IF frequency_drops_below_threshold THEN shed_load IN ConsumerZones
G: Minimize_cost_of_generation AND Minimize_transmission_losses
D: FrequencyStabilityDaemon monitoring grid_frequency

The COH model demonstrates how intelligent infrastructure systems can maintain stability through constraint governance while optimizing performance through adaptive prediction and control.

6.4. Personalized Education: Adaptive Learning Platform

System Description and Significance: This system personalizes educational content and pacing based on individual student characteristics and learning progress. Its significance lies in addressing diverse learning needs within educational settings, potentially improving outcomes through tailored instruction while accommodating different learning styles and paces.

Analysis and Design: The system must model student knowledge states, learning content characteristics, and pedagogical strategies. Identity constraints encode fundamental educational principles (like the zone of proximal development), while neural components adapt to individual learning patterns. The hierarchical structure separates student modeling from content management and instructional strategy.

Modelling Solution:
C: {StudentModel, KnowledgeGraph, ContentBank, TutorAgent}
A: student_knowledge_state, learning_objective, content_difficulty, engagement_level
M: recommend_content(), adjust_difficulty(), generate_quiz()
N: KnowledgeTracingModel, ContentEffectivenessPredictor
E: Embedding of student's cognitive and emotional state relative to learning objective
I: Next content must be within Zone of Proximal Development
T: IF quiz_performance < threshold THEN recommend_remedial_content
G: Maximize_long_term_knowledge_retention AND Maintain_student_engagement
D: FrustrationDaemon monitoring time-on-task and error rates

The COH framework captures the essence of educational intelligence: adapting to individual differences while maintaining pedagogical integrity through carefully designed constraints.

6.5. Industrial IoT: Smart Assembly Line Optimizer

System Description and Significance: This system optimizes manufacturing processes by coordinating robots, conveyors, and quality control mechanisms in real-time. Its significance lies in enabling flexible, efficient production while maintaining quality standards—key requirements for modern manufacturing competitiveness.

Analysis and Design: The system must coordinate physical components while responding to production variations and quality issues. Identity constraints encode safety requirements and quality standards, while neural components handle visual inspection and predictive maintenance. The hierarchical structure mirrors the physical layout of production systems.

Modelling Solution:
C: {RobotArm1, ConveyorBelt, QualityControlCamera, CentralCoordinator}
A: robot_joint_angles, belt_speed, defect_count, throughput
M: adjust_speed(), recalibrate_robot(), flag_defective_unit()
N: VisualQualityInspectionModel, PredictiveMaintenanceModel
E: Embedding of overall line efficiency and product quality trend
I: Line must stop if human detected in restricted safety zone
T: IF defect_rate_increases THEN reduce_belt_speed AND increase_camera_sensitivity
G: Maximize_throughput AND Minimize_defect_rate
D: AnomalyDetectionDaemon monitoring sensor feeds for unusual vibrations

 doi:10.20944/preprints202509.2371.v1

The COH model demonstrates how intelligent manufacturing systems balance efficiency objectives with safety and quality constraints through hierarchical control and adaptive monitoring.

6.6. Financial Trading: Algorithmic Trading Agent

System Description and Significance: Algorithmic trading systems automate financial market decisions using quantitative models and real-time data analysis. These systems are significant for their ability to execute trades at speeds and frequencies impossible for human traders, providing liquidity to markets while requiring sophisticated risk management to prevent catastrophic failures.

Analysis and Design: The system must process vast amounts of market data, assess risk in real-time, and execute trades while adhering to regulatory and risk constraints. Neural components are essential for predicting price movements and classifying market regimes, while identity constraints enforce fundamental risk management principles that cannot be violated.

Modelling Solution:

C: {MarketDataFeed, RiskModel, ExecutionEngine}

A: portfolio_value, asset_prices, volatility, position_size

M: execute_buy_order(), execute_sell_order(), hedge_position()

N: PriceDirectionPredictor, MarketRegimeClassifier

E: Embedding of current market sentiment and regime (bullish, volatile, etc.)

I: Value_at_Risk (VaR) must not exceed predefined limit (risk_limit)

T: IF stop_loss_price is hit THEN execute_sell_order IMMEDIATELY

G: Maximize_risk-adjusted_returns

D: VolatilitySpikeDaemon monitoring real-time volatility to protect against breach of I:risk_limit

The COH model captures the delicate balance between aggressive profit-seeking and conservative risk management through its constraint hierarchy, with neural components providing predictive edge while identity constraints maintaining safety boundaries.

6.7. Cybersecurity: Network Intrusion Detection System (NIDS)

System Description and Significance: NIDS monitor network traffic for malicious activities and policy violations, providing critical protection for organizational infrastructure. Their significance has grown with increasing cyber threats, requiring systems that can detect novel attacks while minimizing false positives that disrupt legitimate business operations.

Analysis and Design: The system must analyze network patterns in real-time, distinguish normal from malicious behavior, and respond appropriately to threats. Neural components enable detection of novel attack patterns through anomaly detection, while identity constraints enforce fundamental security policies. The hierarchical structure allows for coordinated response across network segments.

Modelling Solution:

C: {NetworkSensors, PacketAnalyzer, BehavioralProfiler, ThreatIntelligenceFeed}

A: network_traffic, connection_logs, user_behavior, known_threat_signatures

M: scan_packet(), correlate_events(), block_ip_address(), alert_admin()

N: AnomalyDetectionModel for network flow, MalwareBehaviorClassifier

E: Embedding of overall network security posture and threat level

I: Traffic from blacklisted IP must be blocked (hard_rule)

T: IF lateral_movement_detected THEN isolate_affected_hosts

G: Maximize_detection_of_true_positives AND Minimize_false_positives

D: ZeroDayDaemon detecting correlated low-severity anomalies indicating novel attacks

The COH framework enables the NIDS to balance detection sensitivity with operational continuity, using neural components for adaptive threat recognition while constraint daemons provide meta-monitoring for evolving attack strategies.

6.8. Smart Cities: Adaptive Traffic Light Control

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 September 2025 doi:10.20944/preprints202509.2371.v1

System Description and Significance: Intelligent traffic management systems optimize vehicle flow through coordinated signal timing, reducing congestion and improving urban mobility. These systems are significant for addressing growing urbanization challenges and reducing the economic and environmental costs of traffic congestion.

Analysis and Design: The system must coordinate multiple intersections while responding to real-time traffic conditions and emergency situations. Neural components predict traffic flow patterns, while identity constraints ensure safety by preventing conflicting green signals. The hierarchical structure enables both local optimization and city-wide coordination.

Modelling Solution:

C: {Intersection1, Intersection2, ... TrafficManagementCenter}

A: vehicle_count_per_lane, queue_length, current_light_phase, emergency_vehicle_proximity

M: change_light_phase(), extend_green_time(), create_green_wave()

N: TrafficFlowPredictor, CongestionPropagationModel

E: Embedding of city-wide traffic flow efficiency

I: Conflicting light phases must never be green simultaneously (safety)

T: IF emergency_vehicle_approaches THEN preempt_light_sequence

G: Minimize_average_wait_time AND Minimize_total_congestion

D: GridlockPreventionDaemon monitoring queue lengths to prevent system-wide lock-up

The COH model demonstrates how urban infrastructure can be managed through distributed intelligence with central coordination, using predictive capabilities to anticipate congestion while maintaining absolute safety guarantees.

6.9. Agriculture: Precision Irrigation System

System Description and Significance: Precision agriculture systems optimize resource usage by tailoring irrigation to specific crop needs and environmental conditions. These systems are significant for addressing water scarcity challenges while maintaining agricultural productivity, particularly in regions affected by climate change.

Analysis and Design: The system must monitor soil conditions, predict water requirements, and control irrigation equipment while balancing crop health with resource conservation. Neural components model evapotranspiration and crop response, while identity constraints enforce minimum water requirements for plant survival.

Modelling Solution:

C: {FieldZone1, FieldZone2, ... WeatherStation, WaterReservoir}

A: soil_moisture, weather_forecast, crop_type, growth_stage

M: open_valve(), close_valve(), calculate_water_need()

N: EvapotranspirationModel, YieldPredictionModel

E: Embedding of field's overall water health and crop status

I: Soil moisture must not drop below crop-specific wilting point (plant_health)

T: IF forecast indicates_heavy_rain THEN delay_scheduled_irrigation

G: Maximize_crop_yield AND Minimize_water_usage

D: LeakDetectionDaemon monitoring water flow rates against expected usage

The COH framework enables the irrigation system to make context-aware decisions that balance competing objectives, using predictive models to optimize water usage while maintaining crop health through carefully designed constraints.

6.10. Environmental Monitoring: Wildfire Early Detection System

System Description and Significance: Early wildfire detection systems monitor environmental conditions to identify ignition risks and detect fires in their initial stages, enabling rapid response to prevent catastrophic wildfires. These systems are increasingly significant as climate change increases wildfire frequency and intensity.

 doi:10.20944/preprints202509.2371.v1

Analysis and Design: The system must integrate data from multiple sensor types, distinguish false alarms from actual fires, and predict fire spread patterns. Neural components analyze sensor patterns and predict fire behavior, while identity constraints ensure immediate response to confirmed threats.

Modelling Solution:

C: {SensorTower1, SensorTower2, ..., SatelliteImagery, FireSpreadModel}

A: temperature, humidity, wind_speed_direction, smoke_detection, vegetation_dryness

M: trigger_alarm(), predict_fire_spread(), calculate_evacuation_zones()

N: FireIgnitionRiskModel, FireSpreadSimulator (physics-informed neural network)

E: Embedding of regional fire danger level

I: Alarm must be raised immediately if confirmed ignition detected (immediate_response)

T: IF smoke_detected AND temperature_spike THEN confirm_via_satellite AND trigger_alarm

G: Maximize_early_detection_accuracy AND Minimize_prediction_error_for_spread

D: SensorFaultDaemon monitoring data streams for malfunctioning sensors

The COH model demonstrates how environmental monitoring systems can integrate diverse data sources through neural components while maintaining reliability through constraint-based verification and response coordination.

6.11. Entertainment & Media: Content Recommendation Engine

System Description and Significance: Recommendation systems personalize content discovery for users based on their preferences and behavior patterns. These systems are significant for managing information overload in digital platforms and enhancing user engagement through personalized experiences.

Analysis and Design: The system must model user preferences, content characteristics, and contextual factors while balancing relevance with diversity. Neural components learn user preferences and content relationships, while identity constraints respect user preferences and content restrictions.

Modelling Solution:

C: {UserProfile, ContentCatalog, RecommendationAlgorithm}

A: user_watch_history, content_features, current_context, diversity_score

M: calculate_relevance_score(), generate_ranked_list(), explore_new_genre()

N: CollaborativeFilteringModel, Content-BasedEmbeddingModel

E: Embedding of user's current mood or intent derived from recent activity

I: Recommendations must not include content user has explicitly banned (user_preference)

T: IF user_abandons_three_shows_in_a_row THEN switch_to_exploration_mode

G: Maximize_user_engagement AND Maximize_long-term_satisfaction

D: FilterBubbleDaemon monitoring recommendation diversity to force exploration when needed

The COH framework addresses the classic recommendation challenge of balancing personalization with serendipity, using constraint daemons to prevent filter bubbles while neural components provide sophisticated preference modeling.

6.12. Logistics & Warehousing: Autonomous Warehouse Picking System

System Description and Significance: Automated warehouse systems coordinate robots to fulfill orders efficiently in large-scale distribution centers. These systems are significant for enabling e-commerce scalability and reducing labor costs while improving order accuracy and speed.

Analysis and Design: The system must coordinate multiple robots, manage inventory, and optimize picking routes while ensuring safety and preventing conflicts. Neural components handle object recognition and grasp planning, while identity constraints enforce collision avoidance and operational boundaries.

Modelling Solution:

C: {MobileRobots, StorageShelves, PickingStations, PackingStations, OrderQueue}
A: robot_location, order_items, shelf_inventory, station_congestion
M: navigate_to_shelf(), pick_item(), transport_to_station()
N: ComputerVisionForItemGrasping, MultiAgentPathPlanningAlgorithm
E: Embedding of overall warehouse throughput and order fulfillment status
I: Two robots cannot occupy same physical space (collision_avoidance)
T: IF order_is_priority THEN reassign_robots_to_fulfill_it_first
G: Maximize_orders_fulfilled_per_hour AND Minimize_robot_congestion
D: BottleneckDetectionDaemon identifying slow-moving areas for dynamic reassignment

The COH model demonstrates multi-agent coordination in physical environments, using hierarchical control to scale to large numbers of robots while maintaining safety and efficiency through constraint governance.

6.13. Domestic Robotics: Household Assistant Robot

System Description and Significance: Domestic robots assist with household tasks through physical interaction with human environments. These systems are significant for addressing aging populations and labor shortages while requiring sophisticated human-robot interaction capabilities.

Analysis and Design: The system must perceive unstructured home environments, manipulate objects safely, and understand human commands. Neural components enable perception and natural language understanding, while identity constraints ensure human safety during physical interactions.

Modelling Solution:
C: {NavigationSystem, Manipulator, VisionSystem, TaskPlanner, UserInterface}
A: user_command, room_map, object_locations, current_task
M: recognize_speech(), fetch_object(), clean_room()
N: ObjectRecognitionModel, NaturalLanguageUnderstandingModel, GraspPlanningModel
E: Embedding of household state and user's current needs/intent
I: Robot must not collide with humans or pets (safety)
T: IF user_says "bring me my phone" THEN locate_phone AND plan_path AND pick_up
G: Successfully_complete_user_tasks AND Minimize_energy_usage
D: BatteryDaemon ensuring robot returns to charger before shutdown

The COH framework enables safe and effective human-robot coexistence by combining sophisticated perception and action with rigorous safety constraints and energy management.

6.14. Computational Scientific Discovery: Automated Hypothesis Generation

System Description and Significance: AI systems for scientific discovery analyze vast literature and data to generate novel hypotheses and research directions. These systems are significant for accelerating scientific progress by overcoming human cognitive limitations in processing large-scale scientific information.

Analysis and Design: The system must extract knowledge from scientific literature, identify patterns in experimental data, and generate testable hypotheses consistent with established scientific principles. Neural components enable knowledge extraction and pattern recognition, while identity constraints enforce scientific rigor.

Modelling Solution:
C: {LiteratureCorpus, ExperimentalDataset, HypothesisGenerator, StatisticalTestModule}
A: scientific_papers, data_points, candidate_hypotheses, p_values
M: extract_relationships(), generate_hypothesis(), run_statistical_test()
N: LargeLanguageModel for reading papers, CausalDiscoveryAlgorithm
E: Embedding of current knowledge landscape and promising research directions
I: Hypotheses must be falsifiable and consistent with established laws (scientific_integrity)
T: IF new_dataset_is_ingested THEN generate_new_hypotheses AND test_them
G: Maximize_novelty_and_significance_of_findings

D: ReproducibilityDaemon ensuring hypotheses are tested on held-out data

The COH model formalizes the scientific process itself, using neural components to overcome information overload while constraint governance maintains scientific rigor and reproducibility.

6.15. Human-Computer Interaction: Context-Aware Personal Assistant

System Description and Significance: Intelligent assistants understand user context and intent to provide proactive support across devices and applications. These systems are significant for reducing cognitive load and improving productivity through natural multimodal interaction.

Analysis and Design: The system must understand natural language, model user context, and coordinate across services while respecting privacy and security boundaries. Neural components enable speech recognition and intent understanding, while identity constraints enforce privacy protections.

Modelling Solution:

C: {SpeechRecognizer, DialogueManager, ServiceIntegrators, ContextMonitor}

A: user_query, dialogue_history, location, time, current_activity

M: transcribe_speech(), determine_user_intent(), execute_service_command()

N: Speech-to-TextModel, IntentClassificationModel, ContextualLanguageModel

E: Embedding of user's current context and intent beyond literal query

I: Must not execute commands compromising privacy/security without confirmation (privacy)

T: IF user_asks "remind me to call mom when I get home" THEN set_geofence_trigger

G: Correctly_fulfill_user_intent AND Minimize_conversational_turnaround_time

D: ConfusionDaemon detecting repeated/rephrased queries to trigger clarification

The COH framework enables assistants to understand user needs in context while maintaining appropriate boundaries, using neural components for natural interaction and constraint daemons for meta-dialogue management.

These models of intelligent systems in 15 application domains demonstrate the comprehensive applicability of the COH framework across diverse intelligent systems domains. Each system illustrates how the COH components work together to create adaptive, constrained intelligence suitable for real-world applications.

7. Implementation of COH Models

7.1. From Model to Executable System

Translating COH models into executable systems requires a structured approach that maintains the framework's theoretical integrity while addressing practical implementation concerns. The implementation process begins with instantiating the component hierarchy, typically using object-oriented or component-based software architectures [17]. Each component implements its specified attributes and methods, with careful attention to the interfaces between components to ensure proper information flow and coordination.

Neural components are implemented using appropriate machine learning frameworks [6], with particular attention to integration patterns that allow learned models to interact with symbolic components. The constraint system requires a dedicated constraint management engine that evaluates constraints in proper priority order (identity before trigger before goal) and resolves conflicts when they arise. Constraint daemons are implemented as background processes that continuously monitor system state and invoke corrective actions when constraints are violated or threatened.

7.2. Runtime Architecture and Execution

The runtime architecture for COH-based systems typically follows a layered approach that mirrors the theoretical framework. The base layer manages the component hierarchy and attribute states. The adaptation layer handles neural components and embedding calculations. The constraint layer evaluates constraints and determines appropriate actions. This separation ensures that adaptive learning occurs within a governed framework that maintains system safety and integrity.

Execution follows a constraint-driven cycle where daemons continuously monitor constraint satisfaction, triggers respond to specific events, and goal constraints guide optimization decisions. The embedding component provides a unified representation of system state that supports coherent decision-making across the hierarchy. Implementation frameworks for COH systems often incorporate simulation and testing capabilities that allow constraint behavior to be verified before deployment in critical applications [18].

8. Comparative Analysis with Alternative Frameworks

8.1. Advantages Over Traditional Approaches

COH provides several significant advantages over traditional approaches to intelligent systems design. Compared to purely symbolic approaches [1], COH integrates adaptive neural components that handle uncertainty and learning more effectively. Unlike connectionist approaches [2] that often function as black boxes, COH maintains explainability through its explicit constraint hierarchy and component structure. Compared to hybrid systems [3] that typically combine symbolic and connectionist elements in ad hoc ways, COH provides a principled integration framework with clear semantics for component interactions.

The framework's neuroscience grounding distinguishes it from engineering-focused approaches that lack biological plausibility. This grounding suggests that systems designed using COH may exhibit more human-like flexibility and adaptability. The explicit constraint hierarchy provides mechanisms for ensuring safety and ethical behavior that are often lacking in machine learning systems [19]. The compositional nature of COH supports modular development and verification, addressing key challenges in complex system engineering.

8.2. Relationship to Contemporary AI Paradigms

COH aligns with several contemporary AI paradigms while addressing their limitations. Like hierarchical reinforcement learning [7], COH recognizes the importance of abstraction levels in complex decision-making, but provides a more comprehensive framework for representing and enforcing constraints. Similar to neuro-symbolic AI [10], COH integrates neural and symbolic components, but with a clearer theoretical foundation and more explicit governance mechanisms.

The framework incorporates ideas from cognitive architectures [20] but with greater formal rigor and implementation guidance. Compared to deep learning systems [6] that excel at pattern recognition but struggle with reasoning and constraint satisfaction, COH provides structured reasoning capabilities alongside adaptive learning. The constraint-based approach shares similarities with constraint programming [2] but extends it with adaptive components and hierarchical organization more suited to dynamic environments.

9. Key Contributions

9.1. Theoretical Foundations for AGI

This paper's primary contribution is establishing COH as a comprehensive theoretical framework for AGI that integrates structural, adaptive, and governance aspects of intelligence. The 9-tuple formalism provides a unified language for describing intelligent systems across domains, addressing a critical gap in AGI research. The framework's neuroscience grounding offers biological plausibility while its formal rigor supports engineering implementation.

The constraint hierarchy represents a significant advancement in modelling intelligent behavior, providing explicit mechanisms for safety (identity constraints), reactivity (trigger constraints), and optimization (goal constraints). The concept of constraint daemons introduces a meta-cognitive layer that enables self-monitoring and adaptive control. These theoretical innovations provide a foundation for developing intelligent systems with predictable properties and verifiable behavior.

9.2. Practical Methodologies and Applications

Beyond theoretical contributions, this paper provides practical methodologies for analysing, designing, and implementing intelligent systems using the COH framework. The detailed modelling of 15 systems across diverse domains demonstrates the framework's versatility and provides templates for application in other domains. The implementation guidelines bridge the gap between theoretical modelling and practical system development.

The paper contributes to multiple application domains by showing how domain-specific challenges can be addressed through COH-based design patterns. Each modelled system illustrates how intelligence emerges from the interaction of hierarchical composition, neural adaptation, and constraint governance. These examples provide valuable case studies for researchers and practitioners working in intelligent systems development.

10. Conclusion and Future Research

10.1. Summary of Findings

This paper has presented Constrained Object Hierarchies as a unified framework for modelling intelligent systems across diverse application domains. The COH formalism captures essential aspects of intelligence through its 9-tuple representation, integrating hierarchical composition, adaptive capabilities, and multi-level constraint governance. The framework's neuroscience grounding provides biological plausibility while its formal rigor supports engineering implementation.

Through detailed modelling of 15 intelligent systems, we have demonstrated COH's versatility in capturing domain-specific intelligence requirements while maintaining a consistent theoretical foundation. The framework addresses key challenges in AGI research, including the integration of learning and reasoning, maintenance of safety and integrity, and support for flexible goal-directed behavior. COH provides both a descriptive language for understanding intelligent systems and a prescriptive methodology for designing them.

10.2. Future Research Directions

Future research should explore several promising directions. Formal methods for verifying COH properties would enhance the framework's applicability to safety-critical systems [21]. Development of COH-specific programming languages and tools could streamline implementation. Empirical studies comparing COH-based systems with alternative approaches would validate the framework's practical benefits.

Research is needed on learning mechanisms for constraint adaptation, allowing systems to refine their constraints based on experience while maintaining essential safeguards. Scaling laws for COH systems should be investigated to understand how framework performance changes with system complexity. Finally, cognitive science studies could further validate the neuroscience grounding of COH and identify additional principles from natural intelligence that could inform framework enhancements.

References

1. L. M. Augusto, "From Symbols to Knowledge Systems: A. Newell and H. A. Simon's Contribution to Symbolic AI," AI & Society, vol. 2, no. 1, pp. 29–62, 2021.
2. D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, no. 6088, pp. 533–536, 1986.

3. R. Barták, M. A. Salido, and F. Rossi, "New Trends in Constraint Satisfaction, Planning, and Scheduling: A Survey," Knowledge Engineering Review, 2023.
4. R. Sun and L. Bookman, Computational Architectures Integrating Neural and Symbolic Processes. Springer, 1994.
5. A. Popescu, S. Polat-Erdeniz, A. Felfernig, et al., "An Overview of Machine Learning Techniques in Constraint Solving," Journal of Intelligent Information Systems, vol. 58, pp. 91–118, 2022.
6. M. Minsky, The Society of Mind. Simon and Schuster, 1988.
7. B. Millidge, A. K. Seth, and C. L. Buckley, "Predictive Coding: A Theoretical and Experimental Review," arXiv:2107.12979, 2022.
8. A. Moeini, J. Wang, J. Beck, et al., "A Survey of In-Context Reinforcement Learning," arXiv:2502.07978, 2025.
9. T. Murakami and K. Ohki, "Thalamocortical Circuits for the Formation of Hierarchical Pathways in the Mammalian Visual Cortex," Frontiers in Neural Circuits, 2023.
10. T. Parr, G. Pezzulo, and K. Friston, Active Inference: The Free Energy Principle in Mind, Brain, and Behavior. MIT Press, 2022.
11. B. C. Colelough and W. Regli, "Neuro-Symbolic AI in 2024: A Systematic Review," arXiv:2501.05435, 2025.
12. R. Shah, A. Irpan, A. M. Turner, et al., "An Approach to Technical AGI Safety and Security," arXiv:2504.01849, 2025.
13. J. L. McClelland, D. E. Rumelhart, and G. E. Hinton, "Parallel Distributed Processing: Explorations in the Microstructure of Cognition," MIT Press, 1986.
14. H. A. Simon, "The Architecture of Complexity," Proceedings of the American Philosophical Society, vol. 106, no. 6, pp. 467–482, 1962.
15. E. C. Tolman, "Cognitive Maps in Rats and Men," Psychological Review, vol. 55, no. 4, pp. 189–208, 1948.
16. W. B. Cannon, The Wisdom of the Body. Norton, 1932.
17. R. E. Bellman, Adaptive Control Processes: A Guided Tour. Princeton University Press, 2015.
18. J. Hawkins and S. Blakeslee, On Intelligence. Times Books, 2004.
19. I. Jacobson, Object-Oriented Software Engineering: A Use Case Driven Approach. Pearson Education India, 1993.
20. B. P. Zeigler, Theory of Modeling and Simulation. John Wiley & Sons, 1976.
21. S. Russell, Human Compatible: Artificial Intelligence and the Problem of Control. Viking, 2019.
22. J. E. Laird, The Soar Cognitive Architecture. MIT Press, 2012.
23. E. M. Clarke, O. Grumberg, and D. A. Peled, Model Checking. MIT Press, 1999.