An Atom-Based Cognitive Architecture for World Modeling and Symbolic Reasoning

Miquel M. de Quadras[1], Olga Valls[2]
[1] CEO, Atomian - mquadras@atomian.com
[2] Lead Developer, Atomian - ovalls@atomian.com
Version 1.0 - August 2026

Abstract

Recent advances in artificial intelligence have largely been driven by connectionist models that operate over high-dimensional distributed representations. While these systems have achieved robust performance across a wide range of tasks, their internal representations are typically not designed as explicit, compositional structures.

Symbolic approaches provide explicit structure and interpretability but often lack a unified mechanism for grounding natural language and continuously operating over heterogeneous information.

This gap has led to hybrid approaches that attempt to combine symbolic structure with neural computation, often through intermediate encoding schemes or task-specific representations. But most existing methods still treat symbolic components as either auxiliary representations or external interfaces rather than as the primary substrate of computation.

In this work, we propose a different approach in which symbolic structure is not treated as an external annotation layer, but as the fundamental representational substrate of the system. We introduce a world model based on symbolic atoms, defined as minimal, orthogonal, discrete units of structured representation. These atoms are designed to serve as the basic elements over which all internal operations are performed.

We introduce Atom Modeling Language (AML), a visual formalism for representing and manipulating the proposed world model. AML provides an explicit representation layer that mirrors the internal structure of the system and enables external inspection of atom-based configurations.

To support interaction with natural language, we define a bidirectional interface that maps between natural language and the symbolic atom representation. This interface allows linguistic input to be systematically grounded into structured internal forms and conversely allows structured representations to be expressed in natural language.

Finally, we define an architecture that operates directly over symbolic atoms. The architecture processes language inputs by constructing atomic representations, performs internal transformations

over these structures, and generates natural language outputs from the resulting configurations. This design treats structured symbolic manipulation as the central computational mechanism of the system.

Table of Contents

1. Introduction

This work is not concerned with improving perception systems, learning mechanisms, or neural architectures. It does not propose a new model of data-driven representation, nor does it attempt to optimize existing connectionist approaches. It addresses the representation and manipulation of knowledge once representations have been identified. The mechanisms by which sensory inputs are perceived, recognized, or matched to representations are considered external to the model.

It is also not an extension of symbolic formalisms in the traditional sense. In particular, it is not an ontology engineering framework, nor a knowledge graph representation scheme, nor a rule-based reasoning system built over predefined logical structures.

The objective of this work is narrower and more structural: it investigates the constraints under which knowledge representation can be decomposed into a minimal set of orthogonal primitives without introducing overlapping representational roles. The focus is not on application-specific expressiveness, but on the underlying conditions that make compositional structure possible in a non-redundant form.

In this context, common interpretations of symbolic systems should be explicitly avoided. The proposed framework does not treat symbolic elements as labels over latent representations, nor as syntactic annotations attached to pre-existing semantic content. Likewise, relational constructs should not be interpreted as standard graph edges, and structural mechanisms should not be conflated with linguistic or logical operators from existing formalisms.

The guiding stance of this work is that representational systems are often characterized by implicit entanglement of structural, descriptive, and event-based roles. This work examines the consequences of enforcing strict separation between such roles through orthogonal atomic primitives.

2. Orthogonality Principle

The proposed model is based on the hypothesis that knowledge can be decomposed into a small set of orthogonal atomic functions. Each function captures a distinct role within the system and is defined so as to minimize overlap with the others.

Under this view, complex knowledge structures do not require increasingly specialized mechanisms. Instead, they emerge from the composition of a limited number of primitive atomic forms and their interactions.

The design prioritizes orthogonality over primitive expressiveness, seeking the smallest set of atomic functions capable of accounting for the knowledge phenomena it is intended to describe.

3. The Knowledge Model

3.1. Related work

A range of cognitive architectures and knowledge representation systems define a notion of an “atom” as the minimal unit of computation or representation, but they differ substantially in what is considered primitive and how meaning is grounded.

In OpenCog, an Atom is the fundamental unit of the AtomSpace hypergraph, comprising both nodes and links as first-class elements of representation. Atoms are additionally enriched with cognitive metadata, including truth values and attention-related measures, which directly influence reasoning and resource allocation within the system (Goertzel et al., OpenCog / AtomSpace design literature; PLN and ECAN frameworks). In this formulation, atoms are not purely structural objects but hybrid entities combining representation and cognitive state.

In contrast, the LIDA architecture defines the atomic unit of cognition as the cognitive cycle itself, a recurring temporal process integrating perception, attention, and action selection over approximately 100–200 milliseconds (Franklin, Baars). Here, “atom” refers to an irreducible processing episode rather than a representational element.

ACT-R adopts a different stance, where the minimal unit of declarative knowledge is the chunk, a structured slot-value representation stored in declarative memory and accessed via production rules (Anderson). Similarly, Soar defines working memory elements (WMEs) as atomic symbolic assertions in attribute-value form, manipulated by production systems (Laird, Newell, Rosenbloom). In both cases, atomicity is tied to discrete memory structures rather than process or graph primitives.

The Semantic Web framework (RDF) defines atomic knowledge in terms of triples (subject, predicate, object), which constitute the minimal meaningful assertion in a distributed graph of facts (W3C RDF specification). While simpler in structure, RDF still treats relational assertions as primitive units of meaning, with semantics emerging from graph composition.

In contrast, neural and multimodal learning systems do not define explicit symbolic atoms. Instead, they operate on distributed representations, where discrete tokens (text), patches (vision), or frames (audio) serve primarily as input encodings rather than semantically grounded entities. Meaning is encoded implicitly in vector spaces rather than explicit symbolic structures.

Across these systems, “atomicity” alternates between representational units, memory structures, processes, or encoding artifacts. None explicitly separates semantic entities, linguistic units, and physical signal representations as independent ontological categories.

3.2. Atom

Building on prior approaches to atomic structures in cognitive architectures and knowledge systems, we introduce the atom as the fundamental unit of the knowledge model.

The atom is the basic construct of the model. It provides a uniform primitive for encoding semantic entities, linguistic forms, and perceptual signals within a single substrate.

A key aspect of this model is that atoms are defined independently of their possible realizations. Linguistic expressions, sensory signals, and concepts are not treated as different levels of abstraction of a shared object, but as distinct atoms that may be connected through explicit relational links, which are also atoms.

3.3. Atom Families

Inspired by prior work in cognitive architectures and knowledge representation systems, atoms are organized into families according to their functional role within the model. These families provide a coarse-grained organization of atomic structures without introducing additional layers beyond atoms and their relations. Atom families form a disjoint partition of the atomic space.

1. **Declarative Atoms**. Atoms encoding learned regularities of the environment and structured knowledge about the world.
2. **Lexical Atoms.** Atoms corresponding to linguistic units (not their representations), primitive symbolic artifacts (e.g., numbers, chronological time), and other basic lexical building blocks used in cognition.
3. **Representational Atoms.** Endpoints of processed sensory streams pointing to other atoms. In particular, ASCII encoded streams point to words in Semantic Atoms.
4. **Episodic Atoms.** Atoms encoding events and states corresponding to past occurrences.
5. **Planning Atoms.** Atoms encoding events and configurations corresponding to anticipated or intended future states.
6. **Logical Atoms.** Atoms representing formal logical structures and rule-based transformations.
7. **Mathematical Atoms**. Atoms representing mathematical objects.
8. **Associative Atoms.** Atoms forming unstructured or weakly structured connections between atoms, arising from co-occurrence, similarity, or implicit linkage mechanisms.

3.3.1. Declarative Atoms

Entities. Classes, Instances and Meta-entities

Many theories of cognition, semantic memory, and knowledge representation place identifiable objects, concepts, or entities at the center of structured knowledge. The Entity atom adopts a similar role within the proposed model, representing any distinguishable element that can participate in relations, possess qualifications, or be referenced by other atoms.

An entity is an atom that represents any unique, separable, and identifiable element, whether physical or non-physical.

Like many other knowledge representation models, the proposed model distinguishes between **entity-classes** and **entity-instances**. A class represents a category, type, or abstraction that can be shared by multiple entities. An instance is a particular realization of a class.

Fig. 1. Representation of an Entity, or an entity-class, in the proposed model. Internal class names are commonly used only for clarity.

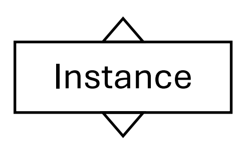


Fig. 2. Representation of an Instance with its internal name.

In addition to classes and instances, the model defines a special kind of entity called a **meta-entity**. A meta-entity is an entity whose purpose is to encapsulate another atom and some resources, allowing them to participate in operations defined for entities. Through the process of meta-entitization, any atom in the model, including classes, instances, relations, qualifications, or other structures, can be represented and manipulated as an entity. The operations associated with meta-entitization are described in the section *Meta-entities*.

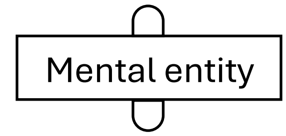


Fig. 3. A Meta-entity.

Modifiers

Modifiers are atoms that represent properties, characteristics, or qualifications that can be associated with other elements of the model.

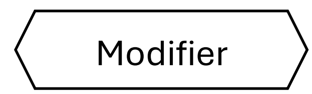


Fig. 4. A Modifier.

While modifiers are atoms themselves, they are most commonly associated with entities, contributing descriptive information about classes or instances. A modifier may represent physical properties, qualities, states, measurements, or any other characteristic relevant to the represented knowledge.

Because modifiers exist as independent atoms, they can be reused across multiple entities and contexts. For example, the same modifier may be associated with a class to express a general characteristic shared by its instances, or with a particular instance to express a specific property of that instance.

Through meta-entitization, modifiers may also be associated with atoms that are not entities. The mechanisms used to establish such associations are described later in this work. Different meta-entities are described in the section *Meta-Entities*.

Actions

Actions are atoms that express change. They normally involve entities and are characterized by dimensional information that describes the nature of the change.

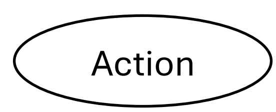


Fig. 5. An Action.

An action may represent changes in position, state, quantity, temperature, or any other measurable dimension. Actions may also involve other actions as part of their definition, allowing complex phenomena to be represented as compositions of simpler changes.

In the proposed model, actions are not primarily defined through taxonomic relationships such as generalization-specialization hierarchies or action entailment structures. Instead, actions are described through their dimensional information (see section *Dimensional Information*), including temporal, spatial, and other relevant dimensions. This information provides sufficient expressive power to distinguish between actions, represent their relationships, and capture different levels of specificity without requiring explicit action taxonomies.

Joints. Relations, Qualifications and Facts.

Joints are atoms that establish structured connections between other atoms in the model. Through joints, entities, modifiers and actions, participate in declarative structures that represent knowledge.

Relations, qualifications, and facts are defined as specific kinds of joints.

Relations are binary joints that connect two entities. The model defines four fundamental relation types:

- Generalization–Specialization. Entity A is a generalization of Entity B, and Entity B is a specialization of Entity A.
- Constituency. Entity B is part of Entity A.
- Identity. Entity A and Entity B represent the same entity.
- Property. Property relations connect entities to specific views of their state. A property is not a generic attribute system, but a constrained relational mapping between entities representing measurable or state-based descriptors.

Fig. 6. A Generalization.

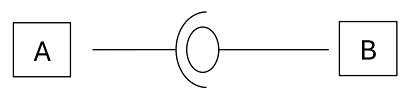


Fig. 7. A Constituency.

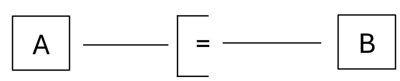


Fig. 8. An Identity.

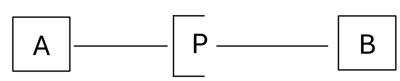


Fig. 9. A Property.

Relations may involve meta-entities through meta-entitization, allowing any atom to be treated as an entity within the relational structure without altering the underlying joint definition.

Qualifications are declarative joints that link an entity to a modifier, specifying characteristics, properties, or descriptive conditions associated with that entity. They define how an entity is described or constrained through the application of a non-instantiable modifier structure.

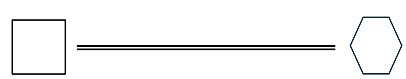


Fig. 10. A Qualification

Facts are action-centered joints that represent events, processes, or state changes. Facts are defined as variable-arity structures centered on an action and associated with a fixed set of roles, with a maximum of four primary roles:

• ACTOR. The entity performing the action.

• OBJECT. The entity directly involved in the action.

• RECEIVER. The entity receiving the outcome of the action.

• TOOL. The entity used to perform the action.

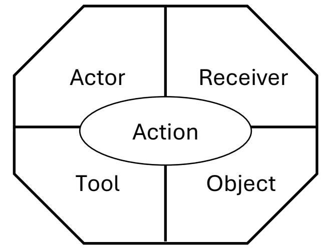


Fig. 11. A Fact with its four roles.

Primitive Resources

The model assumes several primitive grounding resources within the declarative memory: numerical values, temporal references (dates) and characters are some examples. These elements are treated as structurally minimal inputs that may participate in atomic configurations.

Numerical values may be specified or unspecified, and can function either as fixed scalars or as variable placeholders depending on the context of use. They are assumed to span standard numerical domains, including integers, real numbers, and complex numbers. Numerical values may or may not have a unit, which can be entities of any kind.

Temporal references provide a minimal representation of time anchored to discrete points or intervals. Like numerical values, they may be fully specified or partially underspecified, depending on the level of temporal resolution required by the structure in which they participate. Each temporal reference may include structured components, including the day of the week as one of its constituent elements.

Both numerical values and temporal references can be meta-entitized, allowing them to participate in the atomic system as entities when structural operations require it, without altering their role as primitive grounding resources.

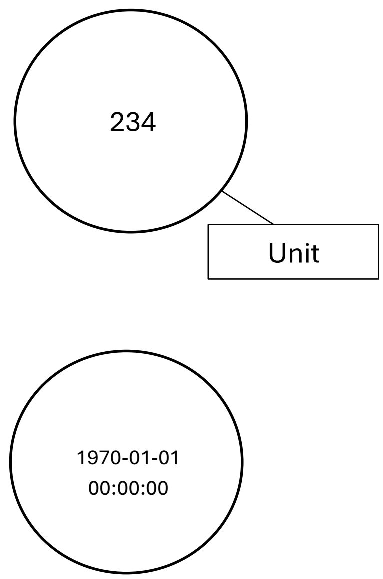


Fig. 12 and 13. Numerical values (with determined values and units) and determined temporal references. Undetermined values and references are represented simply as N and D, respectively.

3.3.2. Lexical Atoms

Lexical atoms correspond to words as minimal units of linguistic form within the model. In contrast to their representational encodings, lexical atoms are defined as abstract linguistic identifiers whose realizations are handled separately within the Representations family of atoms. A single lexical atom may therefore be associated with multiple representational modalities, including handwritten form, character encodings (e.g., ASCII or Unicode), and acoustic realizations.

The distinction between lexical form and representation is consistent with established work in linguistics and semiotics, where signifiers are separated from their physical realizations and may be instantiated across multiple perceptual channels.

Lexical atoms are organized into two broad categories, inspired by the open-class / closed-class distinction in linguistics (cf. traditional lexical category theory in generative grammar and corpus linguistics):

Open-class lexical atoms, including nouns, verbs, adjectives, adverbs, and interjections, corresponding to content-bearing lexical units with high semantic variability.

Closed-class lexical atoms, including prepositions, pronouns, determiners, articles, and modal verbs, functioning primarily as structural operators within linguistic constructions.

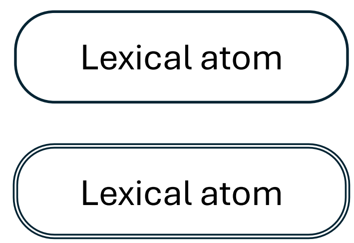


Fig. 14. Open and Closed Lexical Atoms.

Within the model, open-class lexical atoms are associated with other atoms through a primitive weighted grounding association: open-class lexical atoms point to one or more target atoms representing their meanings. This association is not modeled as a joint or relational structure within the atomic system, but is treated as a foundational cross-family correspondence. The mapping is many-to-many, allowing polysemy (one lexical atom associated with multiple target atoms) and synonymy (multiple lexical atoms associated with the same target atom).

Although the primary targets of lexical atoms are declarative atoms, lexical atoms may also point to atoms in other families in specific configurations where cross-family grounding is required. Such cases are not structural relations but remain part of the same primitive correspondence mechanism.

Lexical atoms may share representational encodings with other lexical atoms, a mechanism formally addressed in the Representations family of atoms.

Closed-class lexical atoms are interpreted as linguistic control operators, functioning primarily as structural coordinators over lexical and declarative configurations rather than as content-bearing units.

3.3.3. Representation Atoms

Representation atoms correspond to the physical or symbolic realizations through which other atoms are expressed. They define how an atom is encoded, perceived, transmitted, or externally manifested, but do not contribute additional semantic or structural content.

Fig. 15. A Representation Atom (a letter encoded representation) of a Lexical Atom.

A fundamental distinction of the proposed model is the separation between an atom and its representations. An atom defines a unit of knowledge within the model, whereas a representation atom defines one possible realization of that unit in a particular modality.

Any atom may be associated with one or more representation atoms. Multiple representations may coexist for the same atom, reflecting different forms of expression without altering the identity of the underlying atom. Representation atoms maintain a primitive correspondence with the atoms they realize.

This distinction is particularly important for lexical atoms. A word is not identified with its written form, pronunciation, or digital encoding. Rather, the lexical atom corresponds to the word itself, while handwritten text, ASCII or Unicode strings, printed characters, and acoustic waveforms constitute representations of that lexical atom.

The same principle applies to declarative atoms. An entity corresponding to a particular person is distinct from any photograph, drawing, biometric record, or visual appearance associated with that person. Such elements are representations of the entity rather than the entity itself.

Representation atoms therefore provide the interface between the internal atomic structure of the model and its external manifestations. They allow atoms to be observed, stored, communicated, or rendered across different media while preserving the independence of the underlying atomic structure.

The model does not address the perceptual processes by which representations are recognized, interpreted, or matched to previously known representations. Questions related to visual perception, speech perception, pattern recognition, and similar mechanisms fall outside the scope of the present work. The model assumes that a representation has already been identified and focuses exclusively on its role within the representational structure.

In the case of lexical atoms, this distinction is often less apparent because the correspondence between a representation and a word is typically treated as immediate. Nevertheless, written strings, spoken utterances, and other linguistic forms remain representations of lexical atoms and are not the lexical atoms themselves.

Unlike lexical atoms, representation atoms do not participate in meaning. Unlike declarative atoms, they do not participate directly in the definition of facts, relations, properties, or actions. Their sole purpose is to provide realizations of other atoms in specific representational modalities.

3.3.4. Episodic Atoms

Episodic atoms, or episodes, are concrete realizations of declarative facts (facts). Episodic atoms correspond exclusively to realized occurrences of declarative Facts, representing events that have actually taken place rather than potential, planned, or generalized structures.

Fig. 16. An Episode. It will usually enclose or point to a Fact.

Whereas declarative atoms encode fact structures independent of any specific occurrence, episodic atoms bind such structures to a particular realization. An episode therefore contains a fact together with the specific realizations of its four constituent roles.

Episodic atoms may additionally include dimensional information, particularly temporal coordinates, which situate the occurrence within a defined context. This allows each episode to be anchored to a specific point or interval in time, and when relevant, to other contextual dimensions described in the model (see Section *Dimensional Information*).

Through this structure, episodic atoms preserve both the compositional form of the underlying fact and the uniqueness of its realized occurrence.

3.3.5. Planning Atoms

Planning atoms represent potential future realizations of declarative facts. Like episodic atoms, a planning atom contains a fact together with realizations of its four constituent roles. However, whereas episodic atoms describe realized occurrences, planning atoms describe expected, intended, scheduled, or otherwise anticipated realizations. A planning atom does not imply that the realization will

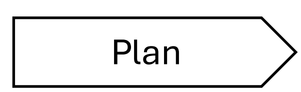


Fig. 17. A Plan. It will usually enclose or point to a Fact.

occur; it merely represents a possible future realization of the corresponding fact. Planning atoms may contain dimensional information, particularly along the temporal axis. See Section *Dimensional Information*.

3.3.6. Logical Atoms

Logical atoms represent explicit logical structures defined over declarative atoms and other logical atoms. They provide a mechanism for expressing structured conditions, dependencies, and compositional statements within the model.

Fig. 18. Simple Proposition (just a rectangle enclosing the atoms). OR, AND and XOR Propositions.

Two categories of logical atoms are defined: propositions and rules.

Propositions represent logical statements. A proposition may directly reference a declarative atom or may combine multiple propositions through logical operators such as conjunction (AND), disjunction (OR), and exclusive disjunction (XOR). Propositions may be recursively composed of other propositions, enabling the construction of arbitrarily complex logical expressions.

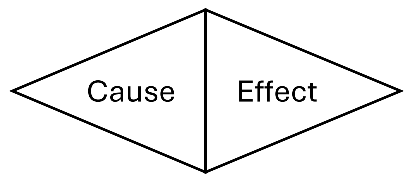


Fig. 19. A Rule. It will enclose or point to Propositions for cause and Effect.

Rules represent directed relationships between propositions. A rule consists of a cause proposition and an effect proposition, expressing a dependency or implication between the two. Both components may themselves be arbitrarily complex propositions.

3.3.7. Mathematical Atoms

Mathematical atoms represent numerical values, arithmetic expressions, and comparison operations. They provide a symbolic mechanism for constructing and evaluating quantitative structures within the model.

Two categories of mathematical atoms are defined: mathematical propositions and comparison operations.

Mathematical Propositions

Mathematical proposition atoms represent numerical values and arithmetic expressions. The following proposition types are currently defined:

Mathematical propositions are represented with the usual corresponding mathematical symbols.

- Numeric propositions
- Addition propositions

- Subtraction propositions
- Multiplication propositions
- Division propositions

Numeric propositions contain the meta-entitization of a numerical value, allowing numbers to participate as structured atoms within the model.

Arithmetic propositions consist of one or more operands and an operator. They are evaluated through sequential composition from left to right. Since operands may themselves be mathematical propositions, arbitrarily complex expressions can be constructed through recursive composition.

This recursive structure allows arithmetic expressions to be represented without introducing additional hierarchical mechanisms.

Additional proposition types, such as exponentiation and root operations, can be incorporated without modifying the underlying atomic structure of the model.

Comparison Operations

Comparison operation atoms represent relations between numerical or arithmetic expressions. Unlike mathematical propositions, comparison operations do not yield numerical values; instead, they evaluate relational conditions between operands.

Mathematical Operations are represented with the usual corresponding mathematical symbols.

The following comparison operations are currently defined:

- Equal to
- Greater than
- Less than
- Greater than or equal to
- Less than or equal to

Comparison operations produce a validity outcome rather than a quantitative result.

3.3.8. Associative Atoms

Associative atoms represent direct pairings between two atoms, capturing links between cognitive elements without structural rules or logical dependencies. They arise from mechanisms such as repeated

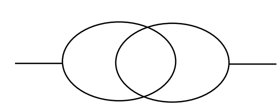


Fig. 20. Associative Atom.

co-activation, contextual proximity, or other processes that establish stable pairings in the model.

An associative atom connects two atoms without introducing additional roles or structural constraints. The relation is symmetric in its simplest form.

Associative atoms may link any type of atom within the system, including declarative, lexical, episodic, and representation atoms. They provide a flexible substrate for similarity, co-occurrence, and memory-driven connectivity that is not captured by other joint types such as logical or relational structures.

3.4. Atomic attributes

Atomic attributes define a fixed set of values attached uniformly to every atom in the model. They do not constitute independent atom types or structural elements, but provide evaluative, dynamic, and descriptive parameters that operate over the atomic substrate.

The model defines five atomic attributes:

3.4.1. Fundamental State (Truth State)

The fundamental state is a binary attribute assigned to atoms, taking the values TRUE or FALSE.

By default, atoms are considered to be in a TRUE state unless explicitly marked as FALSE. The FALSE state represents an explicit override indicating that an atom is negated within the system.

The absence of an atom does not correspond to any truth value. Non-existent atoms are not assigned a state and therefore fall outside the truth evaluation system entirely. Truth is defined only over existing atoms.

This separation ensures that falsity is a structural attribute of an atom, while non-existence is an absence of representation rather than a truth condition.

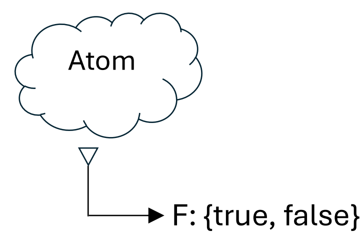


Fig. 21. Fundamental State of an atom.

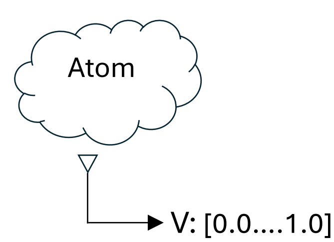


Fig. 22. Veracity Level of an atom.

3.4.2. Veracity Level

The veracity level is a real-valued scalar in the interval [0,1] that expresses the degree of reliability or truth confidence associated with an atom.

It is independent of the fundamental state. Both TRUE and FALSE atoms may have high or low veracity values, forming a two-dimensional evaluative space over atomic state.

This separation allows the model to distinguish between:

- activation (state)
- confidence (veracity)

3.4.3. Reinforcement Level (learning degree)

The reinforcement level is a non-negative integer that represents the accumulated reinforcement of an atom through repeated activation or use.

It encodes the strength or consolidation of an atom within the system's memory dynamics. Higher reinforcement values indicate stronger stabilization over time, while lower values correspond to weaker or less frequently engaged atoms.

This attribute reflects usage history rather than truth evaluation.

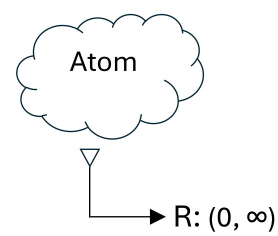


Fig. 23. Reinforcement Level of an atom.

3.4.4.Dimensional Information

Dimensional information (or Story) is a structured atomic attribute that encodes the contextual existence of an atom across temporal, spatial, and scalar domains. It provides a coordinate-based system through which atoms can be situated, compared, and aligned within structured contexts of occurrence.

Unlike scalar-valued attributes, dimensional information is internally structured and composed of dimensional nodes. These nodes define localized positions, intervals, and structural boundaries over each dimension, forming a coherent representation of the atom's contextual footprint within the model.

The model defines three primary dimensional domains:

Time dimension: represents the temporal existence of an atom, including points, intervals, and event boundaries. Temporal structure is inherently ordered and supports sequential interpretation.

Space dimension: represents spatial localization or positional context of an atom. Spatial structure is not inherently ordered and supports relational positioning.

Scalar dimension: represents positioning along quantitative axes such as temperature, intensity, brightness, or any other measurable magnitude.

Each dimensional domain is composed of nodes that represent structured elements such as points, intervals, or boundaries. Nodes may optionally contain other nodes, or reference other nodes or atoms from other dimensions, allowing dimensional structure to be linked with declarative, episodic, or other atomic configurations. Each node has one type from a predefined and well-defined closed set of node types.

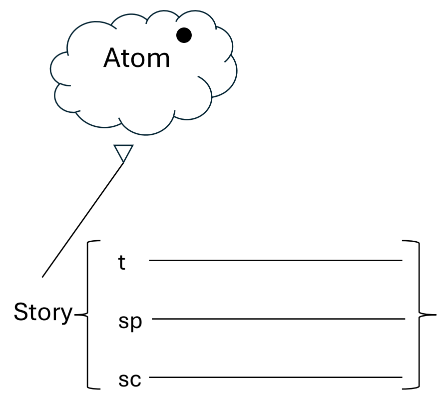


Fig. 24. Dimensional information of an atom.

Through this structure, dimensional information defines a unified "story" of an atom, capturing when, where, and in what quantitative context it is situated within the system.

The specific taxonomy of dimensional nodes and their operational semantics are defined in the architecture specification of the system.

3.4.5.Emotional State

The emotional state is a vector-valued atomic attribute associated with each atom. It is defined as a six-dimensional real-valued vector in $\mathbb{R}^6$, with no bounded range constraints.

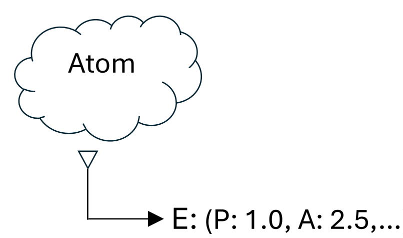


Fig. 25. Emotional State of an atom.

This attribute is intended as a structural parameter for modeling evaluative and affective modulation over atomic configurations.

The model treats emotional states as a primitive evaluative layer that is generated through the activation of atoms. In this sense, the realization of a cognitive configuration produces not only structural outcomes, but also an associated emotional state derived from the participating atoms. Emotional states are therefore interpreted as the aggregate affective modulation induced by atomic activation patterns.

Rather than defining discrete emotion categories, the model represents emotions as continuous vectors across six orthogonal axes. Each axis captures an independent evaluative dimension:

Positiveness: measures the valence of the emotional state along a good–bad axis. Positive values correspond to positive affective valuation, while negative values correspond to negative valuation.

Activation: represents whether the emotional state is energizing (positive values) or deactivating (negative values), influencing the dynamical intensity of cognitive configurations.

Control: captures the degree of perceived regulation or loss of control, distinguishing between controlled (positive) and dysregulated (negative) states.

Extrospection: represents the orientation of attention toward external perception (positive values) versus internal focus or introspection (negative values).

Incompleteness: encodes whether the emotional state is associated with completed or resolved configurations (negative values) or anticipatory and incomplete ones (positive values).

Durability: represents temporal persistence, distinguishing between short-lived, peaked states (positive values) and sustained or slowly decaying states (negative values).

Interpretation

Emotional states do not modify the structural identity of atoms, nor do they participate in relational or logical composition. Instead, they act as a parallel evaluative coordinate system that modulates the interpretation of atomic configurations.

This design allows emotional influence to be represented without introducing categorical emotion types or external behavioral rules, maintaining compatibility with the orthogonal structure of the model.

4. Meta-entitization mechanisms

Meta-entitization is the mechanism through which an existing structure within the model is transformed into an entity without altering its original form. The resulting meta-entity acts as an entity-level representation of the wrapped structure and may participate in relations, qualifications, facts, and other atomic configurations in the same manner as any other entity.

Meta-entitization does not extend the representational primitives of the system; it only promotes existing atomic configurations to entity status, enabling their participation in higher-order atoms without altering their internal structure.

The mechanism is not limited to declarative atoms. Any atom family may be meta-entitized, including logical, mathematical, episodic, planning, lexical, representational, and associative atoms. Primitive resources such as numerical values and temporal references may also be meta-entitized, allowing them to participate directly in the relational structure of the model.

Meta-entitization may additionally be applied to internal structural elements, including fact roles, dimensions, dimensional nodes, and atomic attributes. Through this mechanism, structures that would otherwise remain implicit become explicit objects of representation.

The process of meta-entitization introduces additional structure derived from the wrapped object. For example, when an entity-class is meta-entitized, the resulting meta-entity may be related to the instances belonging to that class. Likewise, when a fact is meta-entitized, its constituent roles may themselves become explicit meta-entities connected to the resulting meta-entity.

A key consequence of meta-entitization is that the model is able to represent and manipulate its own internal structures using the same primitives employed to represent external knowledge. Higher-order constructs therefore do not require additional atom families or specialized representational layers. Instead, they emerge through the recursive application of the same atomic mechanisms defined throughout the model.

Meta-entitization constitutes one of the principal closure mechanisms of the proposed ontology. By allowing arbitrary structures to become entities, it enables the representation of knowledge about knowledge, relations about relations, facts about facts, and, more generally, descriptions of the model's own internal organization using the model itself.

5. Expressive Capacity and Sufficiency of the Model

This section addresses the expressive capacity of the atomic model by combining two complementary perspectives. First, it identifies the internal principles that govern the construction of knowledge within the system. Second, it shows how common knowledge representation and logical formalisms can be systematically expressed within the same structural substrate. Together, these perspectives provide an indirect argument for the sufficiency of the proposed model.

Rather than attempting a formal proof of completeness, the section demonstrates that widely used representational systems can be embedded into the atomic framework without introducing additional primitive constructs.

5.1. Foundational Principles

The expressive capacity of the model is grounded in three core principles: orthogonality, compositionality, and closure.

Orthogonality states that the model decomposes knowledge into independent representational dimensions. Atom families and atomic attributes are designed to capture distinct aspects of knowledge without overlap in function. This separation avoids redundancy in representational roles and ensures that each primitive addresses a unique structural function within the model.

Compositionality states that complex knowledge structures arise from the combination of a fixed set of primitives. Rather than introducing new representational constructs for increasing complexity, the model relies on recursive composition of atoms. This allows arbitrarily complex configurations to emerge from a finite base set.

Closure states that the model does not require the introduction of additional primitive representational mechanisms to achieve higher-order expressiveness. Through meta-entitization, any atomic configuration within the system can be reintroduced as an entity, enabling it to participate in further compositions without modifying its internal nature or introducing new atomic primitives. This mechanism preserves the integrity of the underlying atomic substrate while allowing previously defined configurations to be referenced and combined at higher levels of description. Meta-entitization is applied only once per configuration, ensuring that meta-entities are not recursively re-meta-entitized, thereby preventing unbounded hierarchical escalation while maintaining full expressive reuse of atomic configurations.

Together, these principles define a constrained but expressive representational substrate in which structural diversity emerges from interaction rather than proliferation of primitives.

5.2. Mapping to existing Formalisms

To support the expressive claims of the model, this section outlines how several established knowledge representation and logical formalisms can be expressed within this atomic framework. The purpose of this mapping is not to claim formal equivalence, but to show that these systems can be represented using only the primitives defined in the model.

Semantic networks and knowledge graphs can be expressed through declarative atoms. Nodes correspond to entities, while edges correspond to relations, including generalization, constituency, identity, and property-based connections.

RDF-based representations can be expressed through fact-based configurations, where triples are represented through declarative joints. This provides a direct correspondence between RDF triples and atomic configurations without requiring additional representational mechanisms.

Description logics can be expressed through combinations of declarative atoms, where concepts, roles, and instances are encoded using generalization, qualification, and identity relations. Hierarchical and constraint-based expressions emerge through combinations of these joints.

Logical formalisms, including propositional and first-order systems, can be expressed through logical atoms combined with declarative atoms and relational joints. Propositions correspond to configurations over atoms, while logical dependencies are expressed through relational composition. Higher-order expressions are supported through meta-entitization within the model.

Cyc demonstrates that large-scale knowledge representation systems can be constructed through explicit symbolic decomposition of entities, relations, and rules.

Temporal logic and fuzzy logic can be expressed through atomic attributes attached to logical atoms, allowing temporal qualification, graded evaluation, and variability of truth to be represented within the same atomic system.

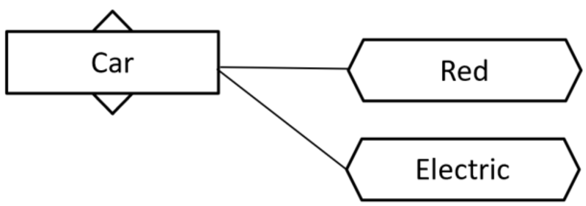
Across these mappings, the key observation is that existing formalisms differ primarily in notation, constraints, and inference procedures, while relying on comparable underlying primitives of entities, relations, and evaluation.

6. AML Examples

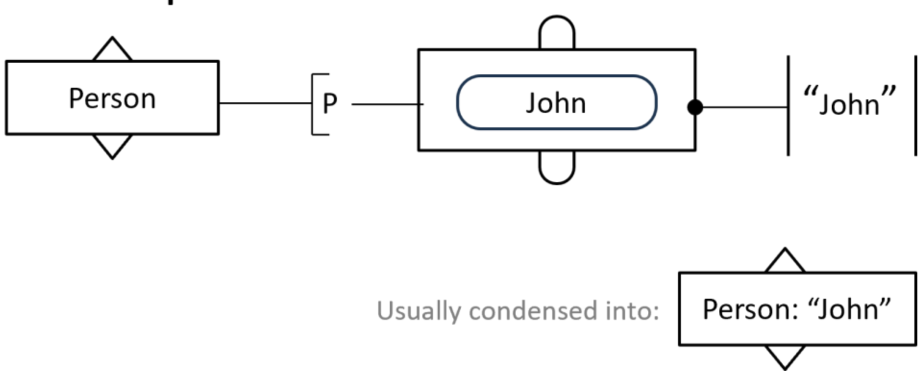
AML (Atom Modeling Language) provides a visual formalism for representing atomic configurations defined by the model. While previous sections define the ontology and expressive mechanisms, AML is used here as a direct projection layer that makes selected configurations observable and inspectable in a compact form.

The purpose of this section is not to introduce new mechanisms, but to illustrate how previously defined atomic families, declarative joints, and atomic attributes combine in concrete cases. Each example increases in complexity, showing how simple atomic relations scale into richer configurations involving multiple atom families, recursive references, and meta-entitized elements.

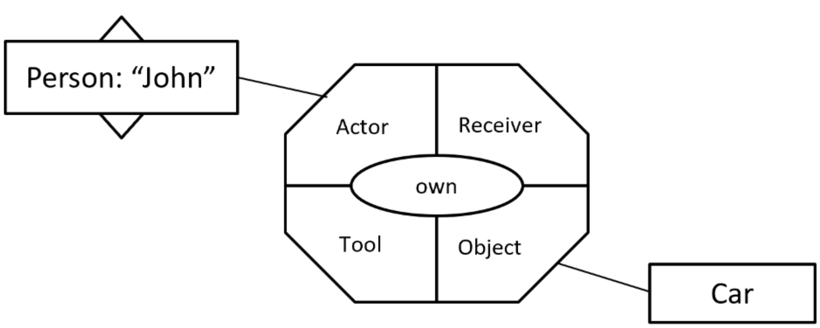
1. A red electric car



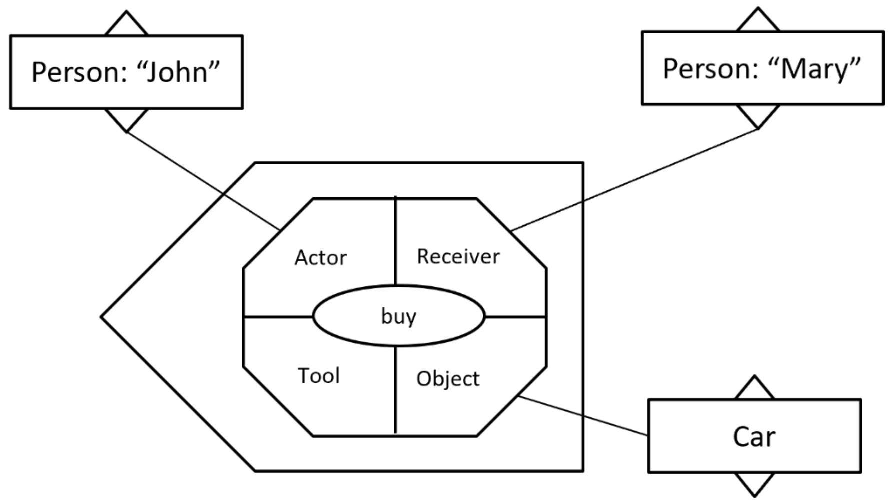
2. John is a person



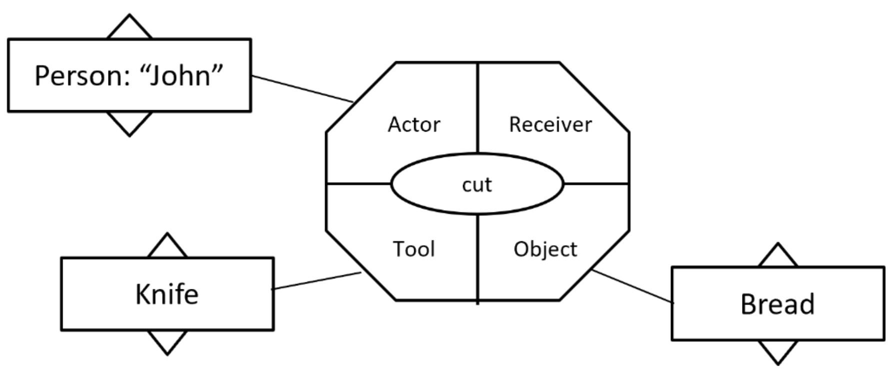
3. John owns a car



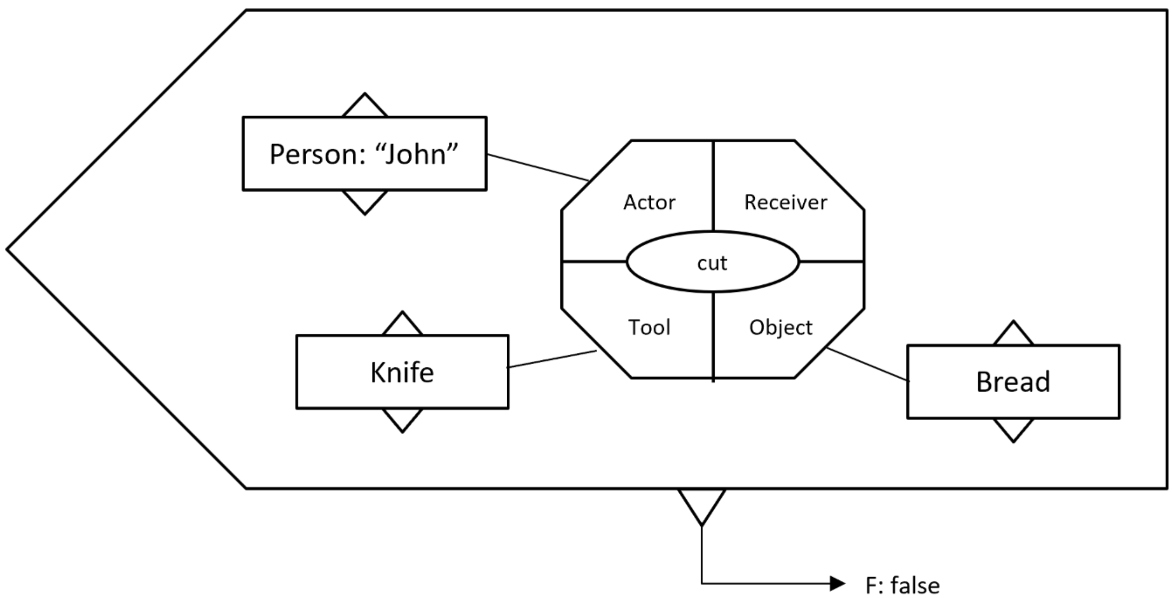
4. John bought the car from Mary



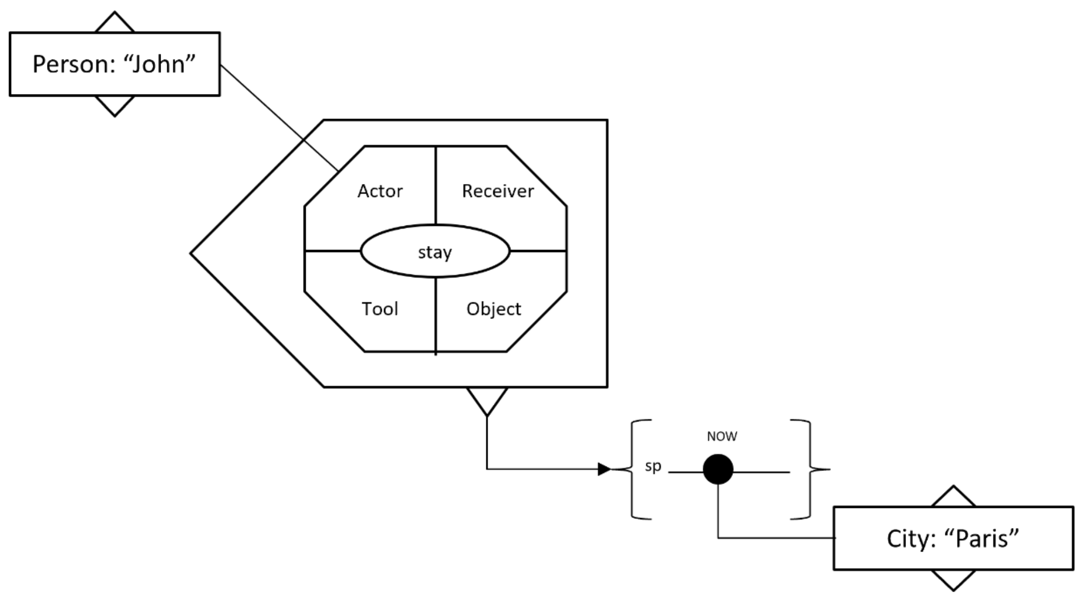
5. John cuts the bread with a knife

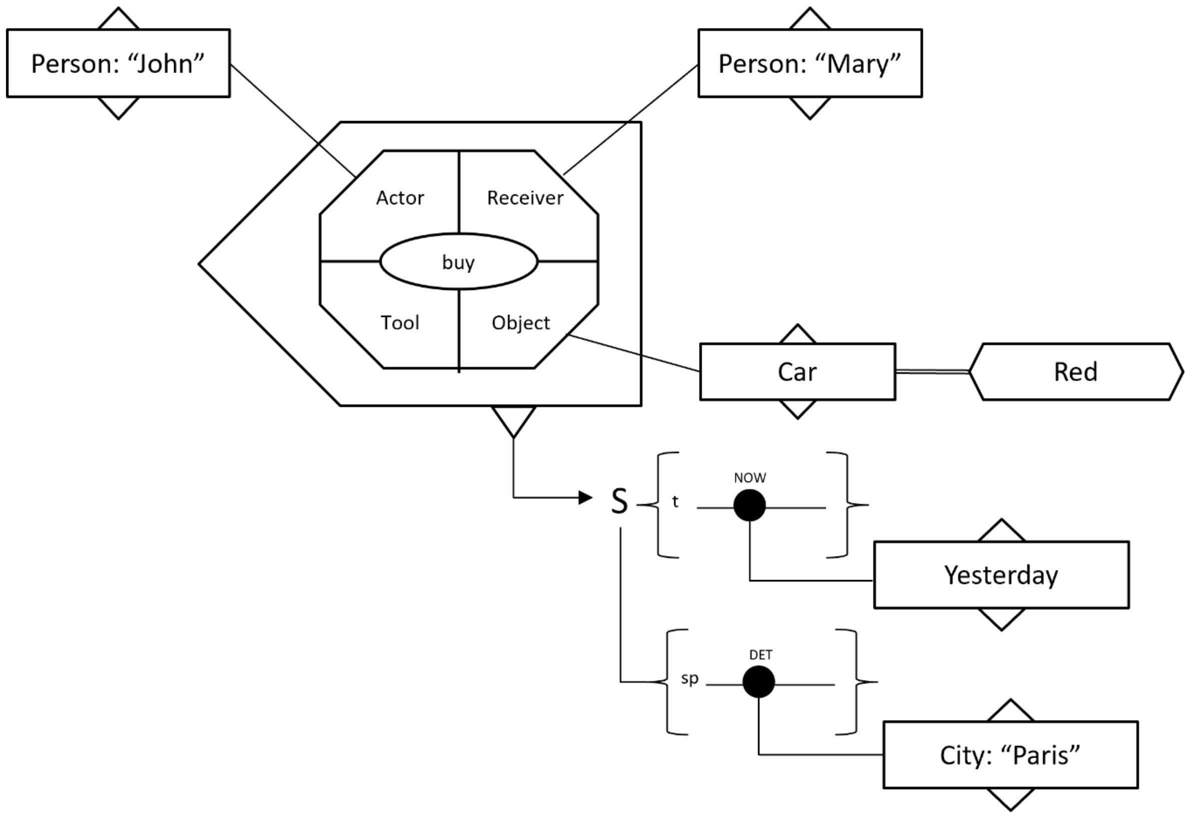


6. John didn't the bread with a knife

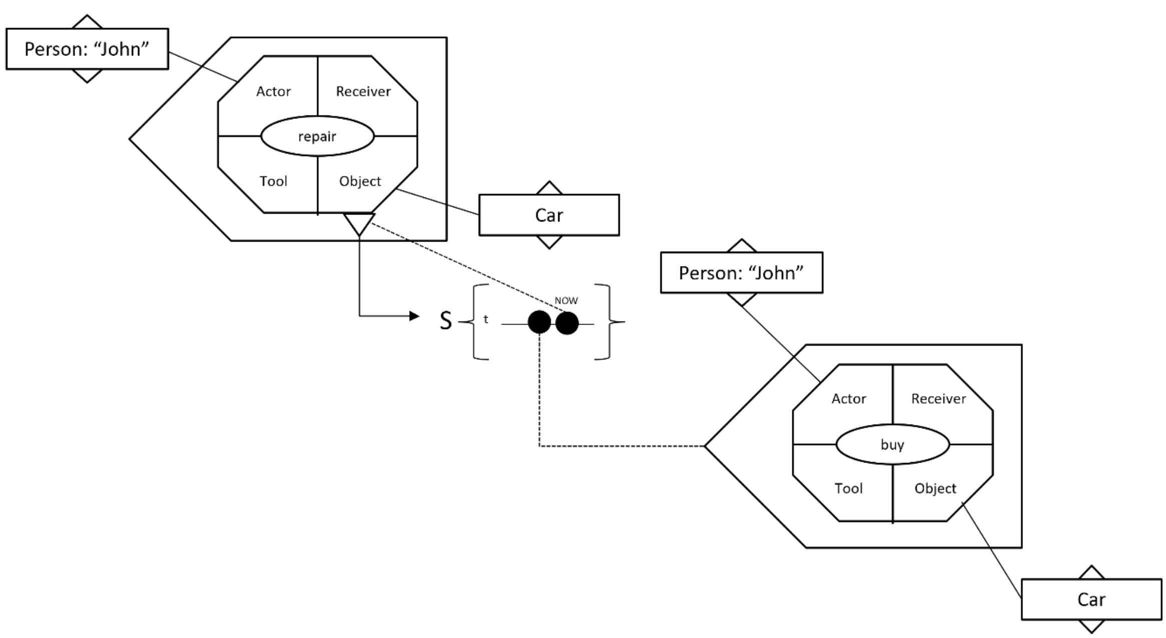


7. John stayed in Paris

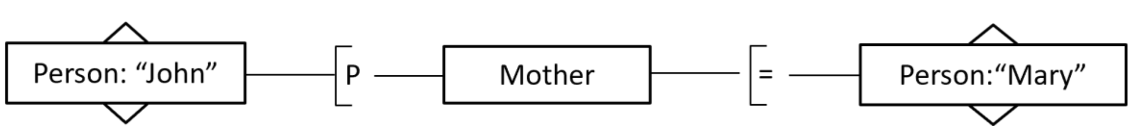




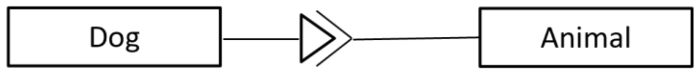
9. John repaired the car after buying it



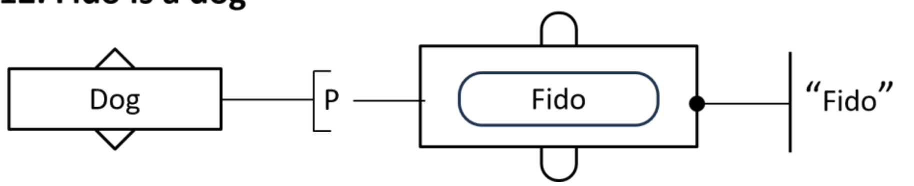
10. Mary is John's mother



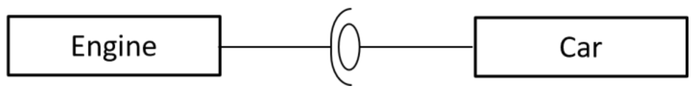
11. Every dog is an animal



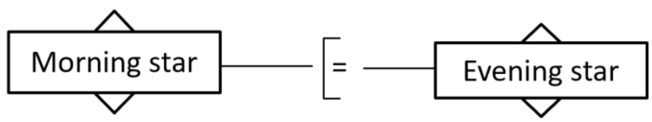
12. Fido is a dog



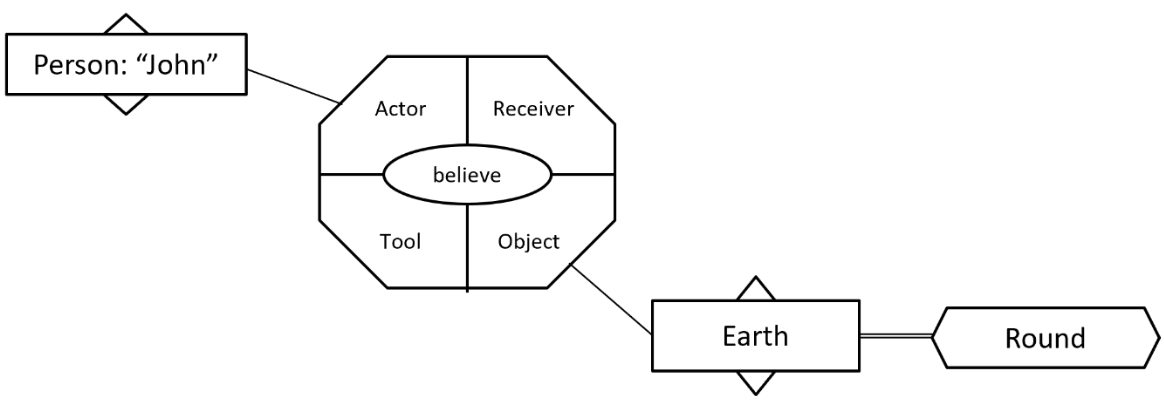
13. The engine is part of the car



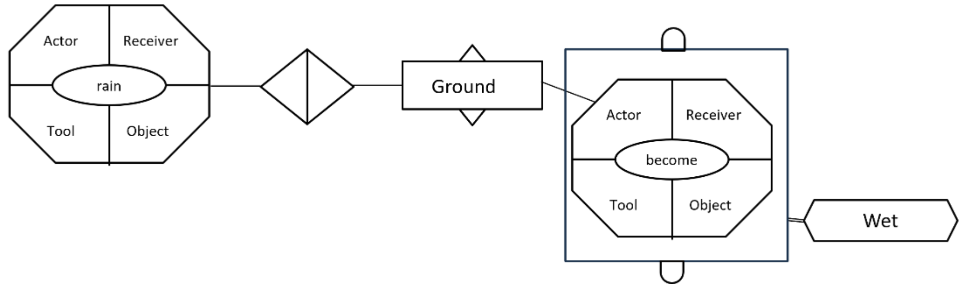
14. The morning star is the evening star



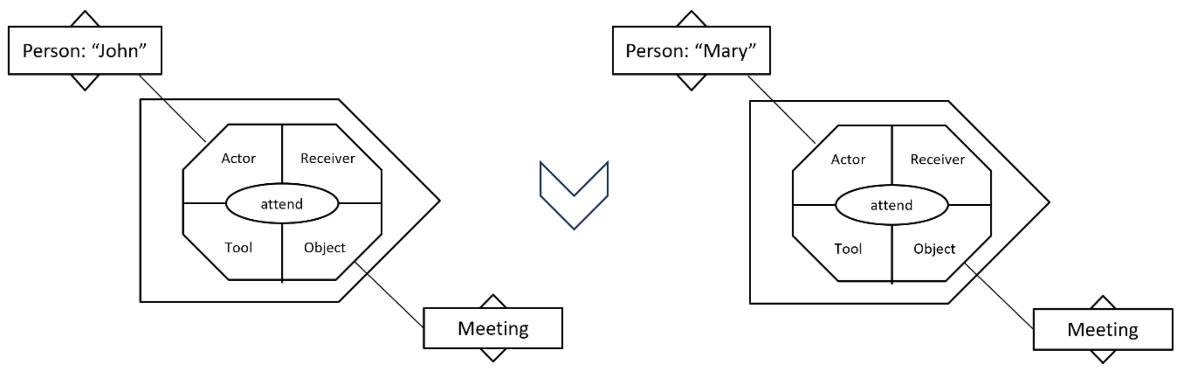
15. John believes that the Earth is round



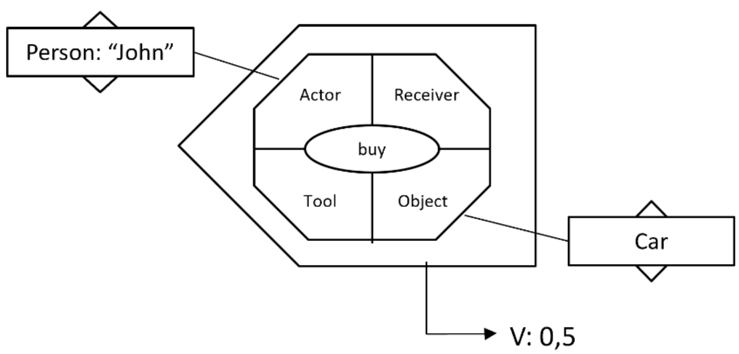
16. If it rains, the ground becomes wet



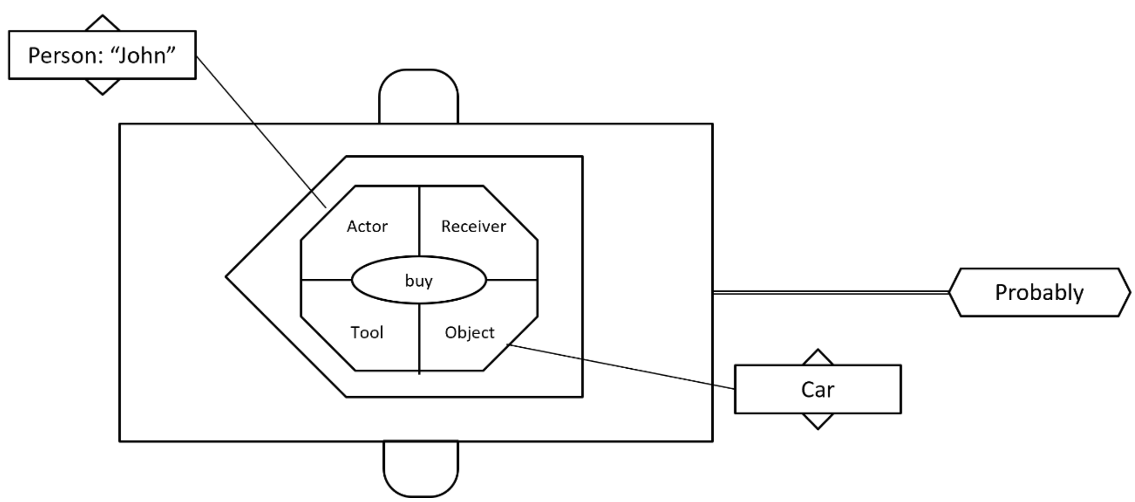
17. Either John or Mary will attend the meeting



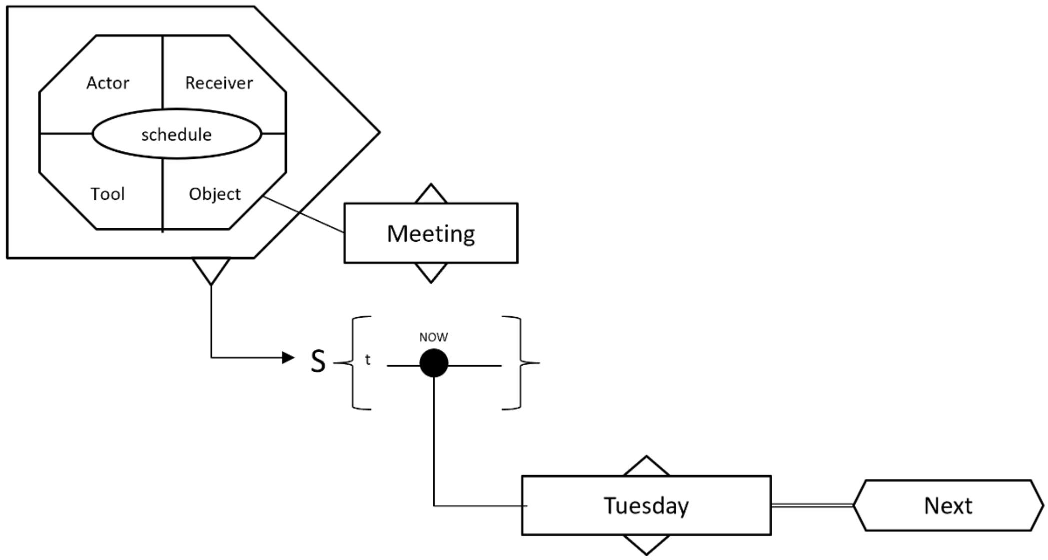
18. John probably bought the car



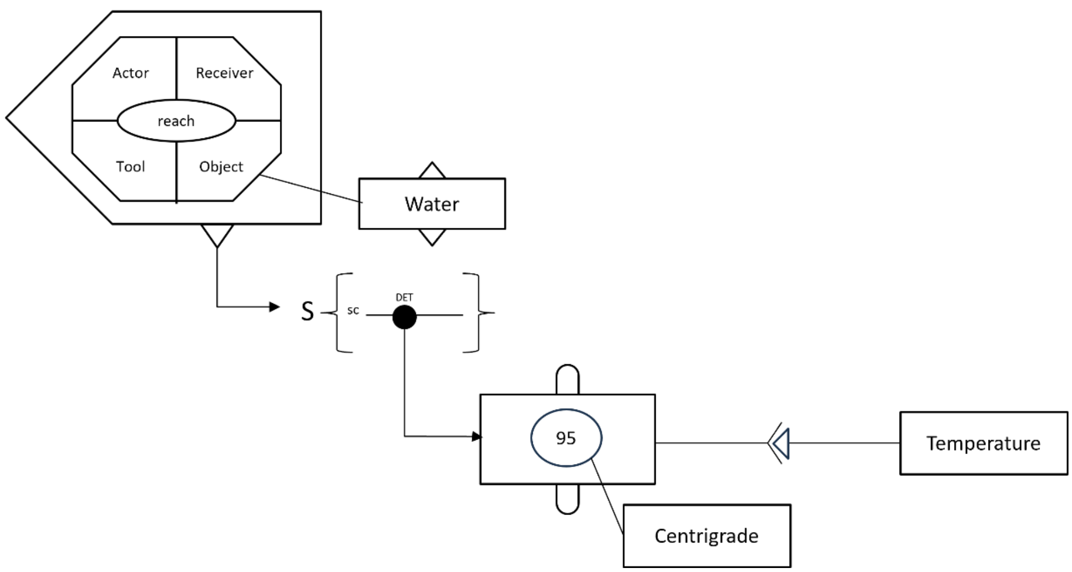
18.b. John probably bought the car (without resolving the adverb)



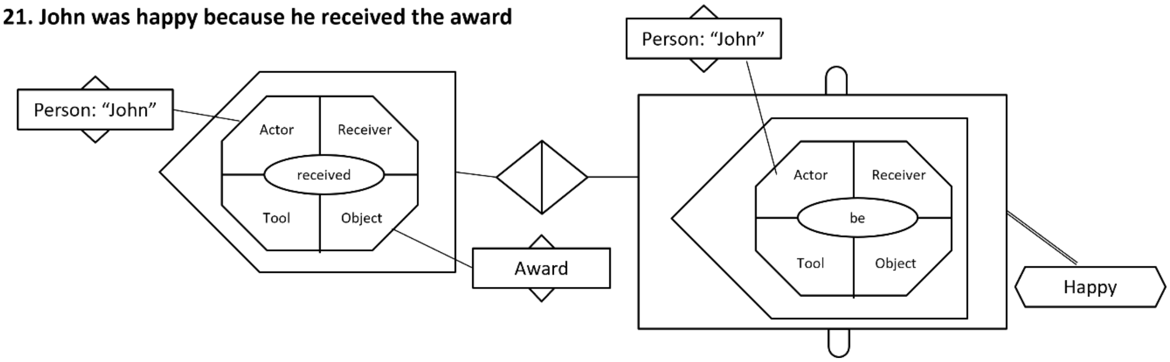
19. The meeting is scheduled for next tuesday



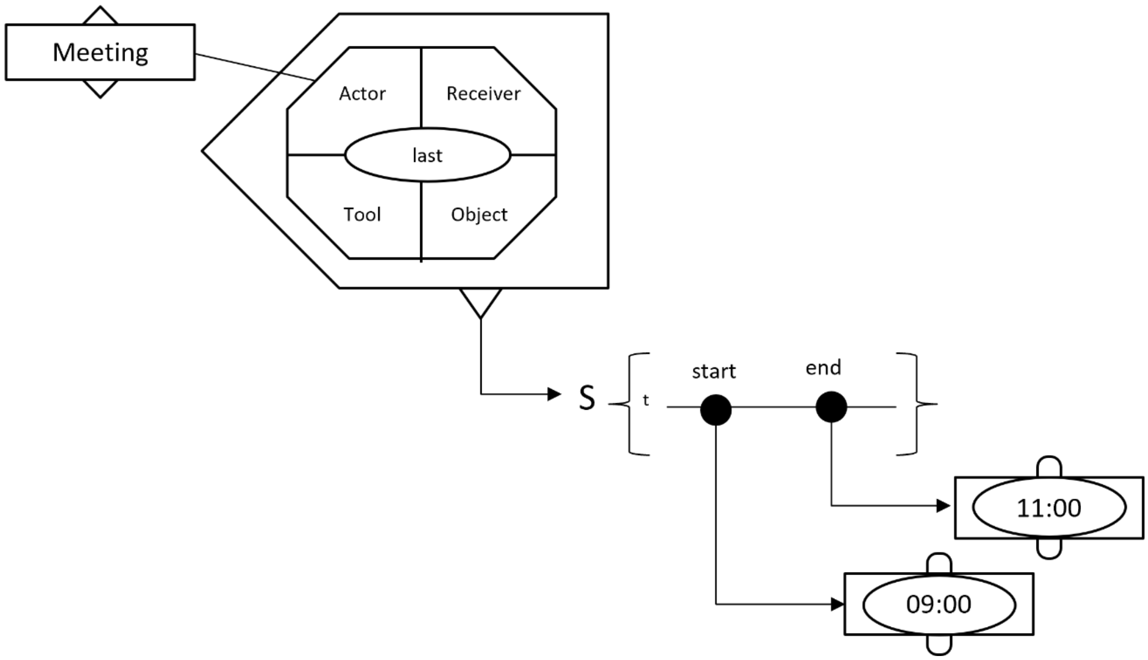
20. The water reached 95ºC



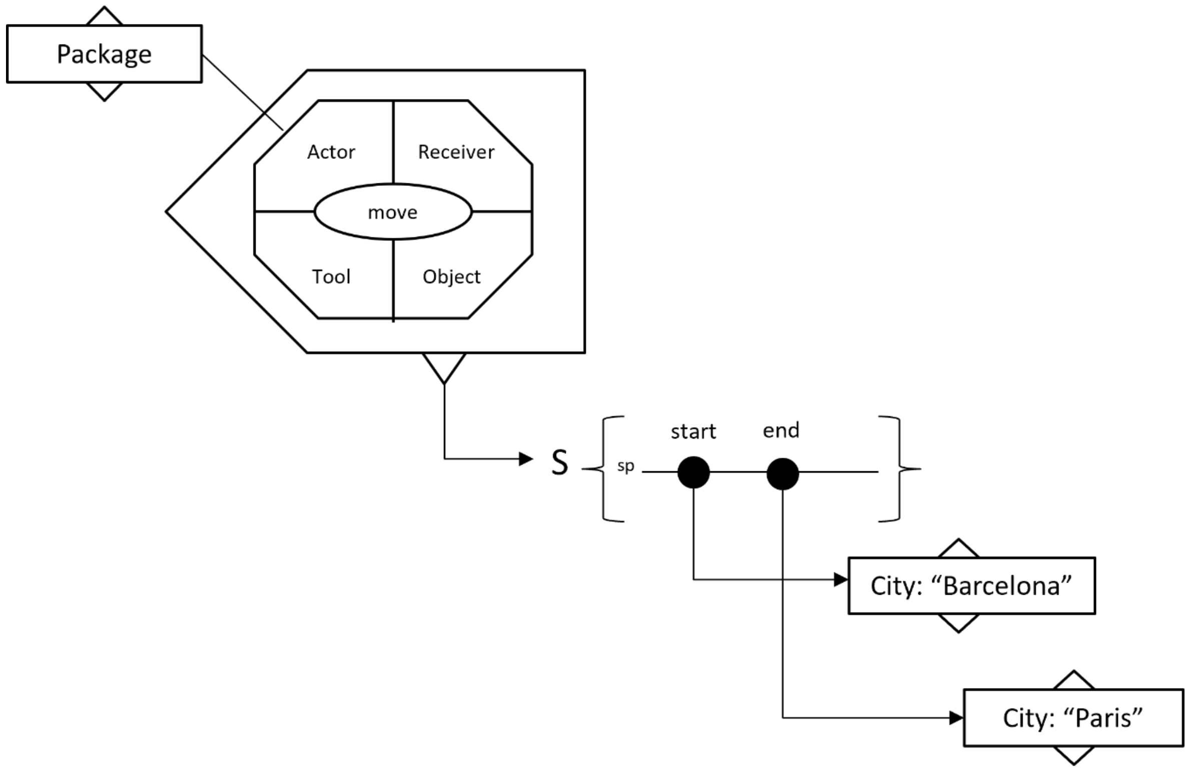
21. John was happy because he received the award



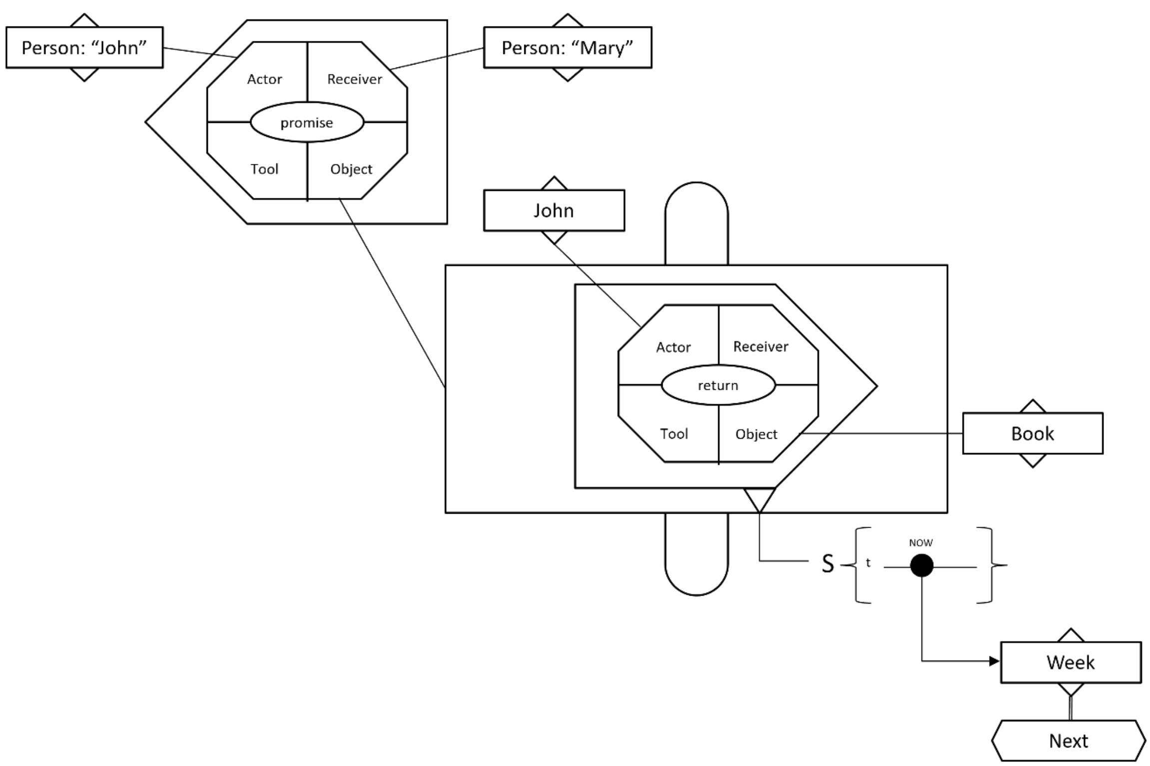
22. The meeting lasted from 9:00 to 11:00



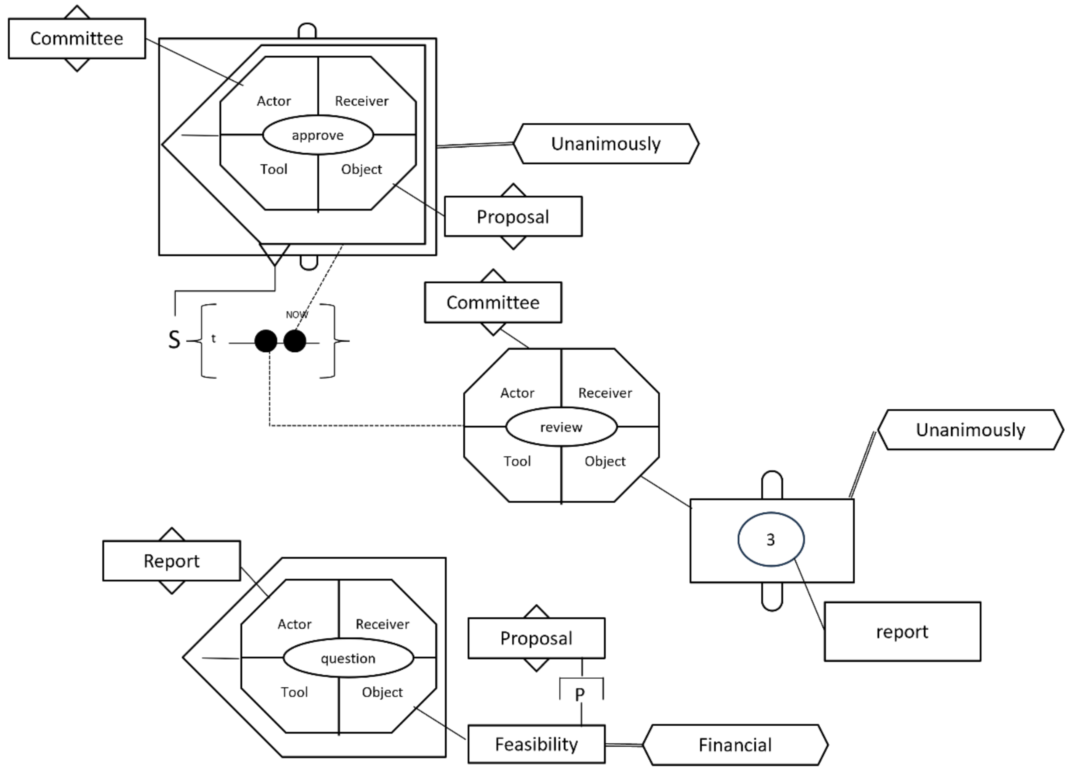
23. The package moved from Barcelona to Paris



24. John promised Mary that he will return the book next week



25. The committee unanimously approved the proposal
after reviewing three independent reports, although one report questioned its financial feasibility



Observations on the examples

These examples also illustrate several properties of the model that are relevant to the interpretation of atomic representations.

First, atomic representation does not require complete interpretation of every atom. An atom may be examined further to determine the meaning or implications of its constituent elements, but this examination is optional and local. For example, the dimensional information associated with *unanimously* may be examined to obtain a more precise account of its meaning, while the atom may also be retained without resolving that meaning further. Similarly, an action such as *approve* may be represented without requiring the model to fully elaborate what approving entails. Not resolving one atom therefore does not prevent the remaining atoms obtained from the same sentence from being retained and used.

Second, much of the meaning associated with a concept is expressed through its dimensional information. This is particularly evident for actions, whose temporal, spatial, or scalar characteristics may provide essential information about how the action is situated. Dimensional information may also contribute significantly to the representation of entities, including their physical characteristics or spatial configuration.

Third, some atoms require contextual resolution to determine their precise reference. Temporal expressions such as *yesterday*, for example, cannot necessarily be reduced to a single absolute temporal value without considering the context in which they occur. The atomic representation can nevertheless preserve the expression and its temporal role without requiring that resolution at the time of its construction.

Fourth, the same meaning may sometimes be represented through more than one atomic configuration. This is not necessarily a contradiction or redundancy in the model, but a consequence of the fact that some semantic relations can be made explicit in different ways. For example, statement *A is a kind of B* may be represented by a fact expressing something that is explicit, or by directly establishing the corresponding generalization relation. These alternatives may ultimately yield equivalent knowledge while differing in their immediate atomic representation.

Finally, some closed-class lexical atoms introduce implications or conventions that are not explicitly expanded into the representation. The conjunction *although*, for example, may indicate that a normally expected condition is not satisfied. Such a convention could in principle be fully codified within the model, but doing so for every lexical item would introduce substantial redundancy and representational cost. In the example, *although* may suggest a rule such as *a proposal receiving a negative review is not accepted*, while the represented facts establish that the proposal was nevertheless accepted. The lexical atom therefore preserves the relevant semantic distinction without requiring the underlying common-sense rule to be explicitly represented.

The examples presented in this section illustrate how the proposed atomic model can represent progressively complex forms of knowledge using a finite set of atomic primitives and their combinations. They also show that representation does not depend on the complete interpretation of every constituent atom, and that contextual resolution and implicit knowledge can remain separate from the atomic representation itself. Together with the principles and mappings discussed above, these examples provide a practical demonstration of the expressive scope of the model.

7. Conclusions

This paper has presented a knowledge representation model based on a finite set of atomic primitives and their combinations. The model provides mechanisms for representing entities, facts, episodes, qualifications, relations, logical content, and dimensional information within a common atomic framework. These mechanisms allow knowledge of substantially different kinds to coexist without requiring distinct representational substrates for each domain.

The expressive capacity of the model arises primarily from the composition of these primitives and from the relationships that can be established between atoms. The principles of **orthogonality**, **compositionality**, and **closure** provide the foundation for this approach. In particular, closure through meta-entitization allows elements of the model itself to become objects of representation, without requiring an unbounded introduction of new primitive mechanisms.

The comparison with established formalisms has shown that concepts found in several important knowledge representation and logical systems can be mapped to the atomic model without introducing additional primitive representational mechanisms. These mappings are not intended to establish formal equivalence with those systems, but to illustrate that the proposed primitives provide a sufficiently general substrate for expressing a broad range of established representational patterns.

The AML examples provide a practical demonstration of this expressive capacity. Progressively more complex examples can be represented through the same atomic mechanisms. The examples also illustrate that complete interpretation of every atom is not a prerequisite for retaining and using the remaining knowledge, and that contextual resolution or implicit knowledge may remain outside the immediate atomic representation when appropriate.

The model presented here should therefore be understood as a representational model, rather than as a complete account of a cognitive architecture. Its implementation, architectural organization, interaction with learning and memory processes, and mechanisms for interfacing atomic representations with natural language are subjects for subsequent work. The present model establishes the underlying representational substrate on which those mechanisms can operate.

8. References

The following works provide foundational background in knowledge representation, logic, linguistics, cognitive modeling, and artificial intelligence relevant to the present model.

Cognitive Architectures and Knowledge Representation

Minsky, M. (1986). *The Society of Mind*. Simon & Schuster.

Laird, J. E., Newell, A., & Rosenbloom, P. S. (1987). *SOAR: An architecture for general intelligence.* Artificial Intelligence.

Newell, A. (1990). *Unified Theories of Cognition*. Harvard University Press.

Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.

Sowa, J. F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley.

Anderson, J. R., Bothell, D., Byrne, M. D., et al. (2004). *An integrated theory of the mind*. Psychological Review.

Semantic Networks, Frames, and Knowledge Structures

Quillian, M. R. (1968). Semantic memory. In *Semantic Information Processing*. MIT Press.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, Plans, Goals, and Understanding*. Lawrence Erlbaum.

Barsalou, L. W. (1992). Frames, concepts, and conceptual fields. In *Frames, Fields, and Contrasts*.

Logic and Formal Systems

Whitehead, A. N., & Russell, B. (1910–1913). Principia Mathematica.

Frege, G. (1879). *Begriffsschrift*.

Gödel, K. (1931). *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I.*

Tarski, A. (1944). The semantic conception of truth. *Philosophy and Phenomenological Research*.

Enderton, H. B. (2001). *A Mathematical Introduction to Logic* (2nd ed.). Academic Press.

Computation and Formal Systems

Shannon, C. E. (1948). *A mathematical theory of communication.*

Turing, A. M. (1936). *On computable numbers, with an application to the Entscheidungsproblem.*

Linguistics and Cognitive Foundations

Chomsky, N. (1957). *Syntactic Structures*. Mouton.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press.

Kucera, H., & Francis, W. N. (1967). *Computational analysis of present-day American English.*

Jackendoff, R. (1990). *Semantic Structures*. MIT Press.

Pinker, S. (1994). *The Language Instinct*. William Morrow.

Miller, G. A. (1995). *WordNet: a lexical database for English.* Communications of the ACM, 38(11), 39–41.

Wierzbicka, A. (1996). *Semantics: Primes and Universals*. Oxford University Press.

Associative and Memory Models

Hebb, D. O. (1949). *The Organization of Behavior*. Wiley.

Anderson, J. R. (1983). *The Architecture of Cognition*. Harvard University Press.

Temporal and Event Representation (foundational work)

Allen, J. F. (1983). *Maintaining knowledge about temporal intervals.* Communications of the ACM.

McCarthy, J., & Hayes, P. J. (1969). *Some philosophical problems from the standpoint of artificial intelligence.* In Machine Intelligence 4.

Knowledge Graphs and Structured Representations

Lenat, D. B., & Guha, R. V. (1989). *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). *Freebase: a collaboratively created graph database for structuring human knowledge.* Proceedings of the ACM SIGMOD International Conference on Management of Data.

Contemporary Cognitive and Hybrid AI Perspectives

Goertzel, B. (2006). *Artificial General Intelligence*. Springer.

Pei Wang (2013). *Non-Axiomatic Logic*: A Model of Intelligent Reasoning. World Scientific.