Atomic Thoughts: A Relational Architecture for Persistent Knowledge in Artificial Intelligence

Toward biologically plausible concept representation and reasoning

Charles Simon 0000-0003-2858-6501
Future AI Society charles@futureaisociety.org

Keywords: Atomic Thoughts, knowledge representation, cognitive architecture, relational reasoning, biologically inspired AI, concept networks, sequence learning, spatial cognition

Abstract

Modern artificial intelligence systems have achieved remarkable success in pattern recognition, language generation, and data analysis. However, most current approaches rely on large numerical parameter sets and lack a persistent internal structure capable of representing evolving knowledge about the world. Biological intelligence, by contrast, appears to maintain stable concepts and relationships that accumulate experience over time [1,7].

This paper proposes the Atomic Thought architecture, a cognitive framework in which knowledge is represented as a persistent relational network composed of small structural elements called Atomic Thoughts. Each Atomic Thought may function both as a concept and as a relationship within the network, allowing complex knowledge structures to emerge from combinations of simple relational units. Concepts, attributes, sequences, and events are all represented using the same primitive structure.

The architecture integrates this knowledge network with an egocentric spatial model representing the system's current environment. Perception inserts recognized concepts into this model, attention selects elements for deeper analysis, prediction generates possible future states, evaluation estimates their effect on system well-being, and decision mechanisms select actions expected to produce favorable outcomes. Learning occurs continuously through a cycle of capture, competition, consolidation, and pruning that reinforces useful structures while removing rarely used ones.

By representing both concepts and relationships using the same primitive element, the Atomic Thought architecture provides a unified framework in which perception, memory, reasoning, and action operate on a shared representation. The resulting system maintains a persistent and evolving knowledge structure capable of supporting recognition, prediction, and adaptive behavior. This approach suggests an alternative architectural direction for artificial intelligence systems that emphasizes relational structure and persistent conceptual representations rather than purely statistical models.

The architecture and knowledge representation described in this paper are illustrated through the figures distributed throughout the text.

1 Introduction

Artificial intelligence has made extraordinary progress in recent years. Large-scale machine learning systems—particularly transformer-based models [5]—have demonstrated impressive capabilities in language generation, image recognition, code synthesis, and many other tasks. By training on vast datasets and scaling model size, these systems have achieved levels of performance that were once thought to be decades away.

Despite these advances, an important gap remains between current AI systems and biological intelligence. Modern AI systems are extremely effective at detecting statistical patterns in data, but they typically do not maintain a persistent internal model of the world. Instead, they reconstruct context from input data and generate outputs based on learned probability distributions. The resulting behavior can appear intelligent, but the underlying mechanism differs fundamentally from how biological systems appear to represent knowledge.

Human cognition is not simply the prediction of the next symbol in a sequence. Biological systems maintain a continuously evolving internal structure in which concepts, relationships, and experiences persist over time. When a person learns that a particular dog named Fido exists, that concept does not disappear after a conversation ends. It becomes part of a growing network of relationships that can be updated, recalled, and combined with new experiences.

This persistence is important. Understanding requires more than recognizing patterns in isolated inputs. It requires maintaining stable concepts, relating those concepts to one another, and updating those relationships as new information is encountered. A cognitive system must be able to connect perception, memory, prediction, and action within a shared internal representation of the world.

The architecture proposed in this paper explores an alternative approach to artificial intelligence built around these principles. Instead of representing knowledge primarily through large numerical parameter sets, the system represents knowledge as a network of small relational structures called Atomic Thoughts. Each Atomic Thought represents a concept or relationship whose meaning arises from its connections to other thoughts in the network.

From these simple primitives, larger cognitive capabilities can emerge. Networks of Atomic Thoughts form a persistent knowledge structure that can store concepts, relationships, sequences, and experiences. When combined with mechanisms for perception, attention, prediction, and evaluation, this relational structure can support the continuous process of interpreting the environment, imagining possible futures, and selecting actions.

The goal of this paper is not to present a complete implementation of a general intelligence system, but rather to describe the architectural principles that such a system may require. In particular, it examines how persistent relational structures can support inheritance, reasoning, prediction, and decision-making within a unified framework.

The remainder of this paper introduces the Atomic Thought representation, explains how relationships and inheritance are expressed within the network, and describes how cognitive processes such as perception, search, prediction, and evaluation can operate on the same underlying structure.

Earlier cognitive architectures such as Soar and related symbolic systems explored persistent symbolic structures for reasoning and memory, though often using representations different from those proposed here [2].

A reference implementation of the Atomic Thought architecture and related experimental systems is available as open-source software on GitHub [11].

1.1 The Architectural Gap

Modern artificial intelligence systems are built on a variety of powerful techniques, including deep neural networks, transformer architectures, reinforcement learning, and knowledge graph systems. Each of these approaches has produced important advances, yet they rely on fundamentally different internal representations of knowledge than those used by biological cognitive systems.

In many current AI systems, knowledge is encoded implicitly in large numerical parameter sets. Neural networks store learned patterns in the weights connecting layers of artificial neurons. While this approach allows extremely complex statistical relationships to be captured, the resulting representations are difficult to interpret and cannot easily represent persistent, identifiable concepts whose relationships evolve over time.

Transformer-based systems illustrate this distinction particularly clearly [5]. These models construct temporary representations of context from input tokens and use them to predict the most probable continuation of a sequence. Once the computation is complete, however, the internal state is discarded. The system does not maintain persistent entities that accumulate experience across interactions. Concepts appear to exist within the generated output, but they are reconstructed dynamically rather than stored as stable structures.

Knowledge graph systems, in contrast, explicitly represent entities and relationships [4]. This approach provides a structured way to represent facts about the world, but such graphs are typically static repositories of information rather than continuously evolving cognitive systems. They store relationships but do not normally support the dynamic competition, reinforcement, and decay processes required for a system that learns continuously from experience.

Reinforcement learning systems introduce the ability to learn behaviors through interaction with an environment [6], yet they typically operate on compressed numerical representations of state rather than explicit conceptual structures. As a result, the knowledge acquired through reinforcement learning is difficult to interpret or reuse outside the specific tasks for which the system was trained.

These architectures have each achieved significant successes within their intended domains. However, they do not provide a unified representation in which concepts, relationships, experiences, and predictions coexist within a persistent internal structure. Instead, perception, memory, reasoning, and decision-making are usually implemented as separate subsystems that operate on different internal representations.

Biological cognition appears to operate differently. The brain maintains stable representations of concepts that persist over time and are continuously updated by new experience. These concepts are connected through a network of relationships that supports recognition, prediction, reasoning, and action within a shared representational framework.

The gap, therefore, is not primarily a matter of computational scale or training data. It is architectural. Current systems excel at pattern recognition and statistical prediction, but they lack a persistent relational structure capable of representing the evolving network of concepts and relationships that appears to underlie biological understanding.

Addressing this gap requires reconsidering the primitive building blocks from which intelligent systems are constructed. If intelligence depends on persistent concepts and relationships that evolve through experience, then those capabilities must be present at the architectural level rather than emerging indirectly from large collections of numerical parameters.

The following sections introduce a primitive representation designed with this objective in mind: the Atomic Thought.

2 Atomic Thought and the Structure of Knowledge

2.1 The Atomic Thought

The Atomic Thought is the fundamental unit of knowledge in the architecture described in this paper. All concepts, attributes, relationships, and sequences are represented using the same primitive structure. By building the system from a single type of element, the architecture maintains a uniform representation for all forms of knowledge.

DEFINITION: Atomic Thought
The smallest unit of relational knowledge, functioning both as a concept and as a Link within the network

The Atomic Thought consists of a few internal parameters and any number of Links to other Atomic Thoughts as shown in Figure 1. An Atomic Thought does not correspond to a traditional database record or a node in a conventional knowledge graph. Instead, it is a small structural element whose meaning emerges from its connections to other Thoughts. A Thought may represent a concept, an attribute, a relationship, or an event depending on how it participates in the network of Links.

In this sense, the role of a Thought is not fixed. The same Atomic Thought may function as a concept in one context and as a component of a relationship in another. Meaning therefore arises from the pattern of connections [3] in which a Thought participates rather than from any intrinsic type assigned to the Thought itself.

For example, the concept dog may appear as the target of one relationship and as the source of another:

[Fido → is-a → dog]
[dog → has → fur]

In these expressions, dog is simply another Atomic Thought participating in multiple Links. The architecture does not require separate structures for entities and relationships; both are represented using the same primitive element.

An Atomic Thought may optionally have a label that allows humans or external systems to refer to it conveniently. Labels serve only as reference identifiers and do not determine the meaning of the Thought. The actual meaning of a Thought emerges entirely from the Links connecting it to other Thoughts. In many cases a Thought may have no label at all if it exists only as part of larger relational structures.

Atomic Thoughts typically represent singular concepts. A Thought refers to a single instance or a single conceptual entity rather than to collections or plural forms. When plural or quantitative meanings are required, they are expressed through additional modifiers rather than by changing the concept itself. For example, the concept dog remains singular, while plurality or quantity can be represented by linking the concept to modifiers such as many, three, or group. In this way the underlying concept remains stable while the network of relationships expresses variations such as number, quantity, or aggregation. This approach preserves the atomic nature of the representation while allowing more complex expressions to emerge through combinations of Atomic Thoughts.

The relationship types that appear within Links are also Atomic Thoughts. For example, the concepts is-a, has, or causes are themselves Thoughts within the network. Because Link Types are Thoughts, they can participate in additional relationships and may themselves have attributes or hierarchical relationships. This allows the system to represent knowledge about relationships using the same primitive structures used for representing concepts.

All knowledge in the system ultimately reduces to networks of Atomic Thoughts connected through Links.

This uniform representation has several advantages. First, it simplifies the structure of the knowledge network. Because all elements are Atomic Thoughts, the system does not need to maintain separate representations for nodes, edges, attributes, or events. Every piece of knowledge is represented using the same basic building block.

Second, the absence of fixed types allows knowledge structures to grow naturally as new relationships are learned. A Thought that initially represents a simple concept may later participate in more complex relational structures without requiring any modification to its internal representation.

Third, the use of a single primitive representation allows higher-order knowledge structures to emerge naturally from combinations of Atomic Thoughts. Complex concepts, conditional relationships, and sequences of events can all be constructed by linking Thoughts together in different patterns.

In this architecture, the meaning of a Thought is therefore determined not by its internal structure but by the Links that connect it to other Thoughts. The following section describes how these Links are formed and how relationships are represented within the network.

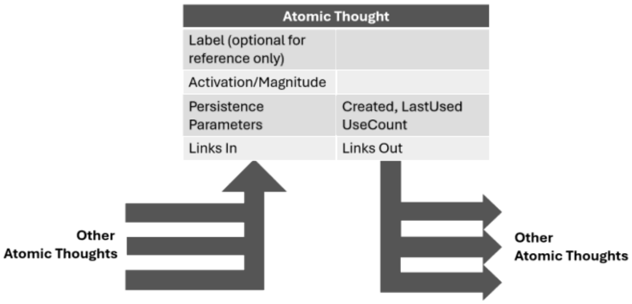


Figure 1. Atomic Thought. An Atomic Thought is the fundamental unit of knowledge in the architecture. Each Thought may optionally contain a label for human reference, an activation value representing its current level of activity or importance, and persistence parameters such as creation time, last use, and use count that influence learning and forgetting. Crucially, every Thought maintains Links to and from other Thoughts, allowing it to participate in the relational network that forms the system's knowledge structure. The meaning of a Thought therefore emerges from the pattern of relationships connecting it to other Thoughts rather than from the internal fields shown in the structure.

2.2 Links as First-Class Structures

In many graph-based knowledge systems, relationships are represented as simple edges connecting nodes. While this representation is useful for storing facts, it places significant limitations on how Links themselves can be described, modified, or reasoned about. Edges typically exist only as structural connectors and cannot easily carry their own properties or participate in additional Links.

The Atomic Thought architecture takes a different approach. In this system, a Link is not merely an edge between two concepts. Instead, each relationship is represented as an Atomic Thought in its own right.

A relationship therefore has the same status as any other concept in the network. It can possess attributes, participate in other Links, and evolve through use just like any other Atomic Thought.

As shown in Figure 2, a basic Link is expressed using the form:

[A → R → B]

where A is the source concept, R is the relationship type, and B is the target concept.

For example:

[Fido → is-a → dog]

In this architecture, the relationship itself is an Atomic Thought that connects the concepts Fido, is-a, and dog. Because this relationship is a Thought, it can participate in additional Links within the network.

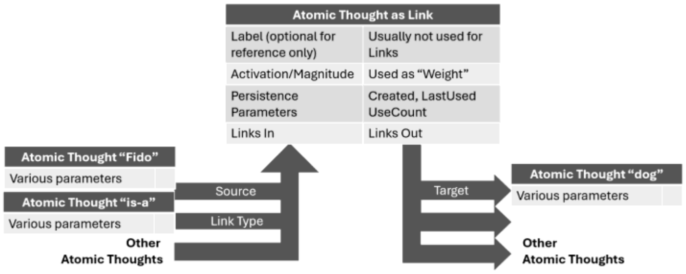


Figure 2. Atomic Thought Acting as a Link. Relationships in the architecture are represented using the same structure as any other Atomic Thought. A Thought functioning as a Link connects a source Thought to a target Thought and identifies the link type, which itself is also represented as an Atomic Thought. Because Links are themselves Thoughts, they can possess the same parameters—such as activation, persistence properties, and additional relationships—allowing relationships to be modified, inherited, or used as components of other Links. This uniform representation enables concepts and relationships to be treated identically within the knowledge network.

For example, a relationship can itself be described, modified, or used within larger structures:

[[Fido → plays → outside] → IF → [weather → is.perhaps → sunny]]

Here, the statement describing Fido playing outside becomes the subject of another Link expressing a condition. Because Links are Atomic Thoughts, the system can represent such higher-order structures without introducing new representation mechanisms.

Treating Links as first-class entities provides several important advantages as shown in Figure 3.

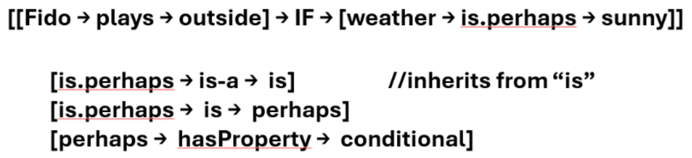
First, Links can carry their own attributes. A relationship may have additional knowledge describing certainty, frequency, temporal context, or other modifiers. Because these properties are expressed using the same relational structure as all other knowledge, no special representation is required.

Second, Links themselves can be organized into conceptual hierarchies. For example, one relationship type may inherit characteristics from another or be classified within a broader category of Links. This allows the system to represent structural knowledge about the Links themselves.

Third, representing Links as Atomic Thoughts allows them to participate in additional Links. Statements can therefore be nested, modified, or conditioned using the same primitive representation. This provides a uniform mechanism for representing facts, attributes, events, conditions, and other higher-level structures.

The result is a knowledge representation in which concepts and relationships are treated uniformly. Rather than distinguishing between nodes and edges as fundamentally different elements, the architecture represents both using the same primitive structure. More complex forms of knowledge emerge naturally from combinations of these simple relational units.

The implications of this design become clearer when examining how the network is traversed to retrieve knowledge and derive inherited attributes. The following section describes how properties attached to relationship types determine the way information propagates through the network during search.



[[Fido → plays → outside] → IF → [weather → is.perhaps → sunny]]

[is.perhaps → is-a → is]   //inherits from "is"
[is.perhaps → is → perhaps]
[perhaps → hasProperty → conditional]

Figure 3. Complex and Conditional Links. Because Links are themselves Atomic Thoughts, they may participate in other relationships, allowing complex logical structures to be represented directly within the network. In this example, the relationship [Fido → plays → outside] becomes the subject of a higher-level Link expressing a condition: the activity occurs IF [weather → is.perhaps → sunny]. By allowing Links to reference other Links, the architecture can represent conditional knowledge, rules, and contextual relationships without introducing a separate rule system. Furthermore, because Link types are also Atomic Thoughts they can incorporate any number of additional attributes themselves.

2.3 Traversal Semantics: Transitivity, Inheritance, and Exclusivity

The Atomic Thought network stores Links directly, without precomputing derived knowledge. Concepts do not permanently inherit attributes from parent concepts. Instead, inheritance emerges dynamically during graph traversal when the system searches for knowledge.

Three properties of link types determine how traversal behaves: transitivity, inheritability, and exclusivity.

2.3.1 Transitive Link Types

Some link types represent Links that can be chained. These link types are marked with the property isTransitive.

When a link with this property is encountered during traversal, the system continues following additional links of the same type.

Example:

[Fido → is-a → dog]
[dog → is-a → mammal]

Because is-a is transitive, a search starting at Fido will continue through the chain:

Fido → dog → mammal

The same behavior applies to certain structural Links. For example:

[person → has → arm]
[arm → has → elbow]

Since "has" is transitive, the traversal continues from arm to elbow.

Transitivity therefore defines how far the search may propagate through a chain of Links.

2.3.2 Inheritable Link Types

Some link types also allow attributes encountered during traversal to be accumulated. These link types are marked with the property isInheritable.

If a link type has this property, the attributes of encountered concepts become candidates for the current concept.

For example:

[Fido → is-a → dog]
[dog → is-a → mammal]
[mammal → has → hair]

Because is-a is both transitive and inheritable, the attributes of mammal become available when searching for the attributes of Fido.

However, not all transitive Links are inheritable. The has relationship is transitive but not inheritable. This distinction allows structural traversal without incorrectly inheriting component properties.

Example:

[person → has → arm]
[arm → has → elbow]
[elbow → can.flex → 150°]

A search starting at person may reach elbow through transitive traversal, but the can.flex property is not inherited as an attribute of person.

This separation between reachability and inheritance allows the system to represent structural Links without propagating inappropriate attributes [3].

2.3.3 Traversal Order

Retrieving attributes from the Atomic Thought network occurs through traversal of Links beginning with the concept(s) of interest. The traversal follows relationships outward through the network, accumulating attributes encountered along the way. Because some Link types are transitive, the search may propagate through chains of related concepts as shown in Figure 4.

For example:

[Fido → is-a → dog]
[dog → is-a → mammal]

If the link type is-a is marked as transitive, traversal beginning at Fido will continue through the chain:

Fido → dog → mammal

As traversal proceeds, attributes associated with each concept may be encountered. If the link type is also marked as inheritable, these attributes become candidate attributes for the concept from which the search began.

For example:

[Fido → is-a → dog]
[dog → can → bark]
[dog → is-a → mammal]
[mammal → is → warm-blooded]

When searching for attributes of Fido, traversal proceeds outward through the network. Attributes attached to dog and mammal are discovered as the traversal reaches those concepts.

Because the is-a relationship is both transitive and inheritable, these attributes are considered attributes of Fido as well.

Traversal therefore determines the scope of inherited knowledge. Transitive link types define how far the search may propagate through the network, while inheritable link types determine which discovered attributes are accumulated during that traversal.

A key advantage of the inheritance mechanism is that many attributes do not need to be stored explicitly for each concept. For example, the fact that Fido can bark may never appear explicitly as a stored Link in the network. Instead, this knowledge can always be regenerated dynamically through traversal of the inheritance chain, such as [Fido → is-a → dog] followed by [dog → can → bark]. This capability is essential for scalability.

A complete description of an individual concept may include hundreds or even thousands of attributes, mostly inherited from more general categories. Storing all of these attributes directly would create enormous redundancy. Instead, the network only needs to store the Links that distinguish a concept from its parents and siblings. All other attributes are recovered automatically through traversal when needed. In effect, inheritance provides a powerful form of data compression in which general knowledge is stored once and reused across many concepts, while individual concepts store only their differentiating properties. This structure also supports efficient search by exception, as described below.

An important property of this process is that traversal requires time. In both computational implementations and biological neural systems, activation propagates step by step across Links. Concepts directly connected to the starting Thought are therefore reached earlier than concepts that require longer chains of Links.

For example:

[Fido → is-a → dog]
[dog → is-a → mammal]
[mammal → is → warm-blooded]

A search beginning at Fido first reaches dog, then mammal, and finally the attributes associated with mammal. This ordering reflects the structure of the network and the propagation of activation through successive Links.

Traversal order therefore plays a central role in determining how inherited knowledge is discovered. Concepts closer to the starting Thought are reached earlier, while more general concepts are encountered later as the traversal continues through the network.

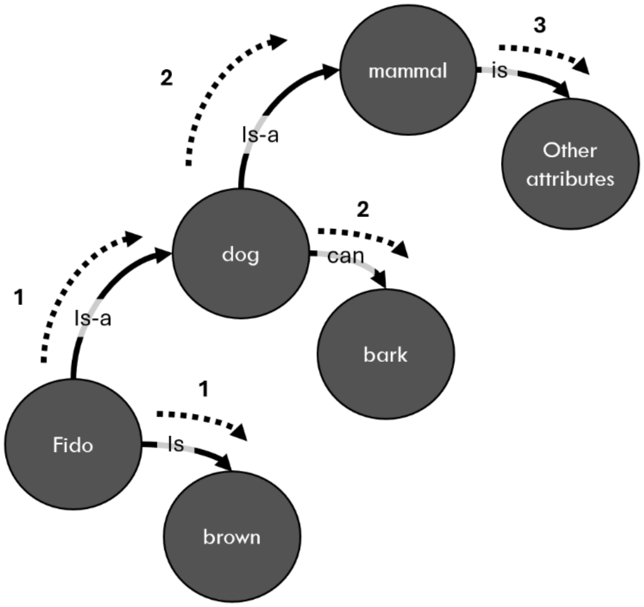


Figure 4. Link Traversal Order During Attribute Retrieval. Attribute retrieval occurs through traversal of the Atomic Thought network beginning with the concept of interest. In this example, traversal begins at Fido (step 1), where attributes directly attached to the concept—such as is → brown—are encountered first. The search then follows transitive is-a relationships to dog (also step 1). In step 2, additional attributes such as can → bark may be discovered. Finally, traversal continues to more general concepts such as mammal (also step 2), where further inherited attributes may be found (step 3). Because traversal requires time, concepts closer to the starting Thought are encountered earlier than those located farther away in the network.

2.3.4 Exclusive Attributes

Some attributes represent mutually exclusive categories. These categories are marked with the property isExclusive.

Example:

[number → hasProperty → isExclusive]

If two candidate attributes belong to an exclusive category, only one may be accepted.

Example:

[dog → has.4 → leg]
[Fido → has.3 → leg]
[Fido → is-a → dog]

When searching for attributes of Fido, the traversal will encounter both Links:

[dog → has.4 → leg]
[Fido → has.3 → leg]

Because the values 3 and 4 are both members of the exclusive category number, only one value may be accepted. The first accepted value (based on the traversal order) blocks conflicting alternatives encountered later during traversal as shown in Figure 5.

2.3.5 Exceptions

This mechanism naturally supports exceptions to general rules.

Inherited attributes from general concepts are encountered later in the traversal than attributes attached directly to the specific concept being examined. When a specific attribute conflicts with an inherited one, the inherited attribute is rejected by the exclusivity rules.

Example:

[dog → has → tail]
[Fido → has.no → tail]
[Fido → is-a → dog]

A search beginning with Fido will encounter:

[Fido → has.no → tail]

before reaching the inherited relationship:

[dog → has → tail]

Because any Link is exclusive of its “not” counterpart, the inherited attribute is rejected. The system therefore represents the exception without removing or modifying the general rule.

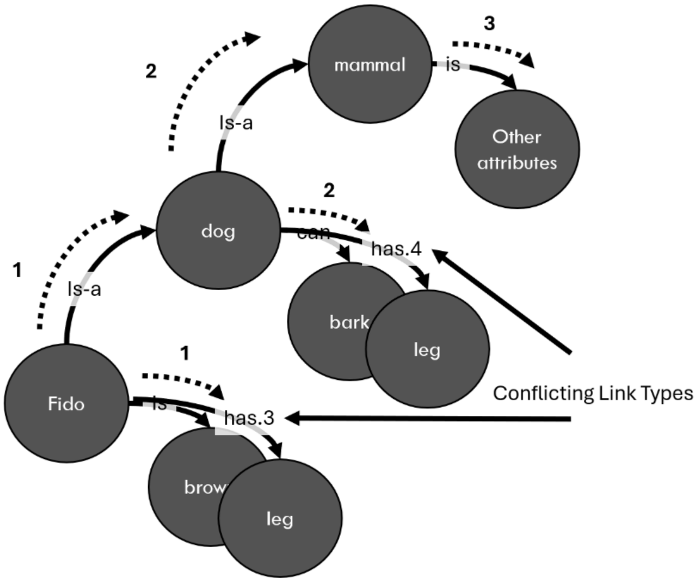


Figure 5. Conflict Resolution Based on Traversal Order. When attributes belong to an exclusive category, conflicting values may arise from different parts of the network. In this example, the specific concept Fido contains the attribute has.3 → leg, while the inherited concept dog contains has.4 → leg. Because traversal reaches the more specific attribute first, the value 3 is accepted and the conflicting inherited value 4 is rejected. This behavior emerges naturally from the traversal process: attributes discovered earlier take precedence over conflicting attributes encountered later.

2.3.6 Multiple Inheritance

A Thought is not limited to a single inheritable or transitive relationship. A concept may inherit attributes from multiple parent concepts simultaneously.

For example:

[Fido → is-a → dog] // inherits attributes associated with dogs
[Fido → is-a → pet] // inherits attributes associated with pets

Because both relationships use the transitive and inheritable link type is-a, attributes from both parent concepts may be accumulated when the network is traversed.

This capability allows the system to represent the fact that real-world concepts often belong to multiple categories simultaneously. A particular entity may inherit behavioral, physical, and social properties from different conceptual hierarchies.

However, when attributes inherited from different parents conflict with one another, the outcome may not be deterministic. Links originating from the same traversal depth do not have a defined ordering, so the order in which competing attributes are encountered during traversal may vary.

For example, two parent concepts provide incompatible attributes (and this is the only available knowledge):

[Suzie → is-a → beautyQueen]
[Suzie → is-a → ditchdigger]

The inherited attributes of these categories may not always be consistent with one another. In such cases the resulting attribute set depends on the traversal path through the network. Conflicts of this type are not uncommon in real knowledge systems and often reflect genuine ambiguity or contradiction in the underlying concepts.

In practice, these inconsistencies may produce a form of conceptual tension or "cognitive dissonance" within the network. Such situations may eventually be resolved through experience, reinforcement, or the addition of more specific attributes attached directly to the concept.

Property	Effect During
isTransitive	Continue following chains of the same link type
isInheritable	Accumulate attributes from encountered concepts
isExclusive	Prevent conflicting attributes from coexisting

Table 1. Summary of properties which impact that influence traversal.

2.4 Sequences

Many forms of knowledge involve events that occur in a particular order. Language, motion, music, and many everyday activities consist of ordered sequences of concepts or actions. The architecture therefore requires a mechanism for representing temporal order.

Sequences are represented using a chain of intermediate Thoughts that represent the individual positions within the sequence as shown in Figure 6. Each sequence step links to the value that appears at that position and to the next step in the sequence. Because the sequence step is a separate structure from the value it contains, the same concept may appear in many different sequences without duplication.

For example, the first step in a sequence containing the letter C might be represented as:

[seq0 → value → C]
[seq0 → next → seq1]

The next step then references the following element in the sequence:

[seq1 → value → A]
[seq1 → next → seq2]
[seq2 → value → T]

In this structure, the sequence itself is represented by a chain of sequence-step Thoughts which results in the spelling of CAT. The values contained in the sequence are connected through separate Links. This separation allows the same concepts to participate in many different sequences and allows subsequences to be shared across larger sequences.

This representation also allows the system to attach attributes like:

[seq1 → timeDelay → 50ms]
[seq1 → probability → 0.3]

without modifying the sequence elements themselves.

Sequences often contain subsequences, and representing these subsequences explicitly provides an important form of compression. In human memory this structure is easy to observe. For example, a melody is typically remembered not as a long string of individual notes but as a collection of recognizable phrases. Likewise, when memorizing speech or text, people often recall clusters of words or phrases rather than isolated elements. Subsequences are represented by the simple expedient of referencing another sequence as the value of a sequence step.

[seq1 → value → otherSeq0]

When this type of reference is encountered, the traversal of a sequence must wait for the traversal of the subsequence to complete. In this manner a complete hierarchy of sequences which reference one another can be created without the need for the complexity of recursion or a structure like call-stacks.

By representing these recurring subsequences as reusable structures, the system avoids storing the same pattern repeatedly. Larger sequences can then reference these subsequences as components, allowing complex patterns to be constructed from smaller, commonly used fragments.

Computers can store sequences in many different ways, including arrays or indexed lists. Biological systems, however, appear to represent sequences through chains of associations [7,9] in which activation propagates from one element to the next. The representation used here follows this biological model. A sequence therefore resembles a linked list in which each element directly references the next.

This representation has several advantages. First, it allows sequences to be traversed naturally as activation spreads through the network. Second, subsequences can be shared among multiple larger sequences. For example, the sequence SET may appear as part of both SETUP and RESET. Because the shared portion of the sequence is represented by the same Atomic Thoughts, it does not need to be duplicated.

As a result, sequences can form branching structures rather than simple linear lists. When activation reaches the end of a shared subsequence, it may continue along the Links of the "calling" sequence OR it can use reverse Links to find all the sequences which reference it. Thus, it allows the system to represent multiple possible continuations of a sequence based on prior experience.

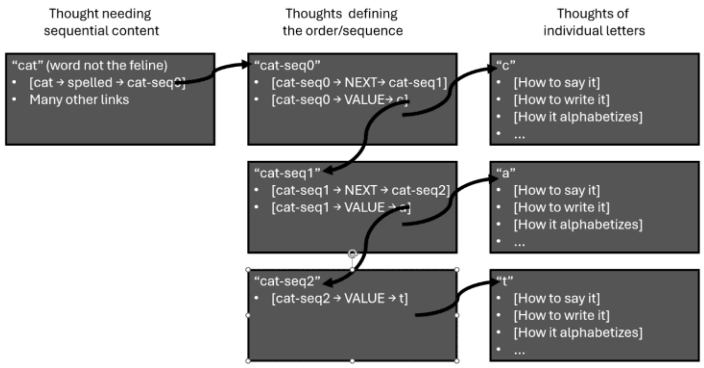


Figure 6. Representation of Sequences Using Dedicated Ordering Thoughts. Sequences are represented using a separate structure that defines ordering independently from the items themselves. In this example, the concept representing the word "cat" links to a sequence structure (cat-seq0, cat-seq1, cat-seq2) that defines the order of elements needed to spell the word through NEXT relationships. Each sequence node references the element it represents using a VALUE link, here pointing to the individual letter Thoughts c, a, and t. Because ordering is stored in dedicated sequence Thoughts rather than in the elements themselves, the same elements can participate in many different sequences without conflict. This allows subsequences and shared components to be reused across multiple contexts while preserving independent ordering relationships.

2.5 Knowledge Search

An intelligent system must be able to retrieve relevant knowledge from its internal structure in order to interpret sensory input, complete partial information, and anticipate future events. In the Atomic Thought architecture, these operations are performed by traversing the network of Links that connect Atomic Thoughts.

Knowledge retrieval occurs continuously during system operation. Rather than relying on a single search mechanism, the system uses three closely related forms of search that operate on the same relational structure. These searches differ primarily in the type of information used to initiate the traversal and the type of result expected.

2.5.1 Attribute Search

The most common form of search occurs when the system attempts to identify a concept based on observed attributes. Sensory input typically provides partial information about an object or event. These observed attributes become the starting points for a search through the network.

For example, a visual perception system might detect attributes such as shape, color, or motion. Each of these attributes corresponds to an Atomic Thought. The system searches for other Atomic Thoughts connected to those attributes through Links that represent possible interpretations.

Conceptually, the process can be viewed as asking the question: What concept best matches the attributes currently being observed?

As the search proceeds, candidate concepts accumulate supporting attributes encountered through the network. Concepts that gather sufficient supporting evidence become likely interpretations of the observed input.

This form of search corresponds closely to recognition processes in biological perception, where partial sensory information activates candidate concepts that compete to explain the input.

2.5.2 Attribute Completion

Once a concept has been identified, the system often needs to retrieve additional information about it. This process is called attribute completion.

When a Thought becomes active, the network is traversed outward through its Links to discover related attributes. These attributes are accumulated during traversal according to the traversal rules described earlier.

For example:

[Fido → is-a → dog]
[dog → has → fur]
[dog → has → tail]

After identifying the concept Fido, a search through the network reveals attributes associated with dog, allowing the system to infer that Fido likely has fur and a tail.

Attribute completion allows the system to fill in missing details about known concepts. This capability is essential for prediction, reasoning, and planning, since real-world observations rarely provide complete information.

2.5.3 Sequence Search

A third form of search operates on sequences of events rather than static attributes. Experiences are often stored as sequences of Links that describe how situations evolve over time.

Because sequence steps are represented as separate structures that reference their values, it is possible to locate all sequences containing a particular element by following reverse value Links. These reverse value Links identify all sequence elements that reference the Thought.

Once these sequence elements are located, the system can traverse the sequence structure. This enables two important capabilities.

First, the system can determine whether additional elements in the sequence match other parts of an observed pattern. Results from multiple sequences gain weight based on the accuracy of the match and so can compete to be the best result.

Second, the system can continue traversing the sequence to discover possible continuations. If a sequence element matches the current situation, following the next Links reveals elements that have historically followed it. These elements therefore represent predictions about what may occur next.

Because a Thought may participate in many sequences simultaneously, this search process can reveal multiple possible continuations. Prediction therefore emerges naturally from the traversal of sequence structures that were previously learned through experience.

2.5.4 Continuous Operation

These three forms of search operate continuously and often simultaneously. Attribute search helps interpret sensory input, attribute completion fills in missing knowledge, and sequence search anticipates future events.

Because all three forms of retrieval rely on the same network of Atomic Thoughts and Links, knowledge stored in one context can immediately influence recognition, prediction, and reasoning in another. This shared structure allows perception, memory, and anticipation to interact within a unified framework.

The following sections describe how these retrieval mechanisms interact with the system's representation of the current environment and how they support prediction, evaluation, and action.

2.6 Language and Concepts

A key distinction in the Atomic Thought architecture is the separation between concepts and the words used to represent them. Words are not the concepts themselves. Instead, words are symbolic structures that refer to concepts within the knowledge network.

In the architecture, a concept such as the dog named Fido is represented by an Atomic Thought that participates in relationships describing its properties and experiences. For example:

[Fido → is-a → dog]
[Fido → is → brown]

These relationships describe the concept itself and exist independently of any particular language.

Words are represented as separate Atomic Thoughts that refer to the underlying concept. A word may therefore be expressed as a relationship between a word symbol and the concept it denotes:

[w:fido → means → Fido]

The prefix w: indicates that the Thought represents a word rather than an abstract concept. Word Thoughts may also contain attributes describing their linguistic properties:

[w:fido → spelled → "fido"]
[w:fido → pronounced → /faɪdoʊ/]
[w:fido → is-a → properName]

Because words are represented as ordinary Atomic Thoughts, they can possess their own attributes and participate in relationships just like any other concept.

2.6.1 Many-to-Many Mapping Between Words and Concepts

Separating words from concepts naturally produces a many-to-many mapping between language and meaning. A single concept may be associated with multiple words, and a single word may refer to different concepts depending on context.

For example, the concept dog may be connected to words in multiple languages:

[w:dog → means → dog]
[w:perro → means → dog]
[w:chien → means → dog]

Words themselves may also be classified according to the language in which they appear:

[w:dog → is-a → EnglishWord]
[w:perro → is-a → SpanishWord]
[w:chien → is-a → FrenchWord]

Because these relationships are expressed using the same relational structure as all other knowledge, the architecture can easily support multiple languages without requiring separate language-specific systems. All words ultimately refer to the same underlying conceptual network.

2.6.2 Phrases as Conceptual Word Units

While many words correspond directly to individual concepts, natural language also contains short phrases whose meaning functions as a single semantic unit. Expressions such as "New York," "hot dog," or "prime minister" convey meanings that are not easily interpreted by considering each word independently. In such cases the phrase itself behaves much like a single lexical item.

The Atomic Thought architecture represents these phrases in the same way it represents individual words: as word Thoughts that refer to underlying concepts.

For example:

[w:hot_dog → means → hotDogFood]

Although the phrase contains two words, the meaning corresponds to a specific concept rather than to the literal combination of the concepts hot and dog. Treating the phrase as a word-level Thought allows the system to associate the correct concept directly with the expression.

Similarly, multi-word names can be represented as single lexical units:

[w:new_york → means → NewYorkCity]
[w:new_york → means → NewYorkState]

Which also illustrates how individual words often have multiple meanings which need to be resolved via the traversal process. In many cases a phrase Thought may coexist with the sequence structure describing its component words:

[w:new → next → w:york]

Because both structures exist within the same network, the system can recognize the phrase either through direct association or through sequence recognition. Once the phrase has been recognized, the phrase Thought can activate the concept(s) it represents.

This approach reflects the way human language processing appears to operate. Frequently encountered phrases become stored and recognized as single units, allowing them to be processed efficiently without reconstructing their meaning from individual words each time they appear.

DEFINITION: Atomic Thought Architecture
A cognitive architecture in which knowledge is represented as a persistent network of Atomic Thoughts whose relationships form the structure of understanding.

By allowing phrases to function as word Thoughts, the architecture supports idioms, compound nouns, and proper names without introducing special mechanisms. Words and phrases simply become different granularities of the same lexical structure, both linking language expressions to the underlying conceptual network.

2.6.3 Word Types and Linguistic Structure

Words can also possess grammatical properties that help guide language interpretation and generation. For example:

[w:dog → is-a → noun]
[w:run → is-a → verb]

These classifications allow the system to learn patterns describing how words tend to appear together in phrases and sentences. Such patterns can be represented using the sequence structures described earlier.

For example, sequences of word types may capture common phrase structures:

[determiner → followed-by → adjective]
[adjective → followed-by → noun]

Similarly, sequences of specific words or phrases may represent commonly observed expressions. Because sequence structures are part of the same knowledge network, language patterns can be learned through experience just like any other sequence of events.

2.6.4 Translation Between Language and Meaning

When language is received as input, the system first interprets the words by following Links from the corresponding word Thoughts to the concepts they represent. These concepts then participate in the broader knowledge network, where their relationships to other concepts provide the meaning of the statement.

When producing language output, the process operates in the reverse direction. Concepts that become active within the spatial model or knowledge network can be mapped to corresponding word Thoughts, allowing the system to express those concepts using the appropriate vocabulary.

In this way, language becomes an interface to the conceptual knowledge network rather than the structure in which knowledge itself is stored. Meaning resides in the network of Atomic Thoughts and their relationships, while words provide symbolic labels that allow the system to communicate that meaning.

3 Cognitive Architecture

3.1 The Knowledge Network

The Atomic Thought architecture represents knowledge as a continuously evolving network of Atomic Thoughts connected by Links. Rather than storing information in fixed structures, the network grows and reorganizes itself as the system encounters new experiences. Concepts, attributes, Links, and sequences all coexist within this shared structure.

A fundamental constraint on any real intelligence is that, at the moment information is received, it is rarely possible to determine whether that information will later prove important. Sensory input arrives continuously, and the significance of a particular observation may only become clear after additional events occur. Because of this uncertainty, a cognitive system cannot rely on selecting only "important" information during perception. Instead, it must initially capture as much of the incoming structure as possible and allow later experience to determine what should be retained.

Biological systems appear to operate in this manner. Experiences are first recorded in a relatively broad and temporary form, after which repeated use determines which elements persist. The Atomic Thought architecture follows this same principle. New observations are initially incorporated into the network with minimal filtering, and the structure of the network is refined over time through reinforcement and pruning.

Learning in this system can therefore be understood as a repeating cycle consisting of four stages as further explained in Figure 7:

capture → compete → consolidate → prune

capture → compete → consolidate → prune

Figure 7. Learning Cycle: Capture → Compete → Consolidate → Prune. Learning occurs through a continuous cycle in which new experiences are first captured as transient structures within the system. These candidate structures then compete for persistence based on relevance, repetition, and their contribution to successful predictions or outcomes. Structures that prove useful are consolidated into the long-term knowledge network, strengthening their persistence and connections. Finally, rarely used or low-value structures are pruned, allowing the system to maintain efficiency while preserving knowledge that contributes to effective behavior.

3.1.1 Capture

When the system encounters new information, the relevant concepts and relationships are added to the network as Atomic Thoughts and Links. This process may occur through perception, through language, or through internal inference.

At this stage the system does not attempt to determine whether the new information represents a reliable or recurring pattern. The immediate goal is simply to preserve the structure of the experience so that it can later participate in recognition, prediction, or reasoning.

For example, observing a short sequence of events may produce Links such as:

[A → followed-by → B]
[B → followed-by → C]

These Links become part of the network even if the sequence occurs only once.

3.1.2 Compete

As the network accumulates experience, multiple Links may describe similar situations or competing interpretations of the same concept. These alternatives naturally compete as the system searches the network during recognition and prediction.

Competition occurs because Links differ in their reinforcement and usage history. Links that are frequently encountered or repeatedly used during search become more strongly represented in the network, while weaker alternatives contribute less influence.

This process allows the network to evaluate competing interpretations through use rather than through explicit rule selection.

3.1.3 Consolidate

Links that are repeatedly reinforced gradually become stable components of the knowledge network. Sequences that occur frequently become reliable predictors of future events, and attributes that consistently appear with a concept become strong descriptors of that concept.

Through this consolidation process, the system extracts stable patterns from accumulated experience. Importantly, this organization emerges gradually through repeated use rather than through explicit instruction.

3.1.4 Prune

Links that are rarely used gradually lose influence within the network. If a Link is not reinforced over time, its persistence decreases and it may eventually be removed from the network.

Pruning is essential because the system initially captures far more information than it ultimately retains. Many observations correspond to temporary circumstances, noise, or one-time events that do not generalize to future situations. Removing these weak Links allows the network to remain efficient while preserving the patterns that truly characterize the environment.

3.1.5 Emergent Structure

Through the continuous cycle of capture, competition, consolidation, and pruning, the knowledge network adapts to the environment over time. Frequently observed patterns become stable components of the network, while transient or irrelevant Links gradually disappear.

This process allows the system to learn from experience without requiring centralized control or explicit rule construction. Concepts, attributes, and sequences organize themselves through repeated interaction with the environment.

Because the same network of Atomic Thoughts and Links supports recognition, prediction, and reasoning, knowledge acquired in one context can immediately influence behavior in another. The structure of the network therefore becomes a continuously evolving representation of the system's accumulated experience.

The following section describes how this knowledge network interacts with the system's representation of the current environment, forming a dynamic internal model that supports perception, prediction, and action.

3.2 The Spatial Model

While the knowledge network stores general knowledge about concepts, relationships, and sequences, an intelligent system must also maintain a representation of the current situation. At any moment the system must know what objects are present, where they are located, and how they relate to one another within the immediate environment.

In the Atomic Thought architecture this function is performed by the Spatial Model. The spatial model represents the system's current understanding of the immediate environment and serves as the workspace in which perception, prediction, and evaluation interact.

Sensory input is first processed by specialized recognition systems. These systems identify candidate concepts from the incoming signals by searching the knowledge network for matches to the observed attributes. The resulting interpretations are then placed into the spatial model, where they become part of the system's current situational representation.

For example, a visual recognition process might identify an object corresponding to the concept dog. Once recognized, that concept is inserted into the spatial model along with information about its location and orientation within the observed environment. The spatial model therefore contains spatial cells describing how objects are arranged relative to one another and Links to the corresponding thoughts in the knowledge network.

The spatial model can be understood as a set of local cells or regions representing the environment surrounding the system. Concepts detected through perception are associated with these regions, allowing the system to represent the positions of objects and events in space. Links between objects—such as proximity, containment, or motion—can then be represented through Links connecting the corresponding Atomic Thoughts.

As shown in Figure 8, The spatial model is organized around an egocentric coordinate system centered on the observing agent. The horizontal axis spans $[-180^\circ, 180^\circ]$ with 0 directly ahead and both -180° and 180° directly behind while the vertical axis spans $[-90^\circ, 90^\circ]$. Unlike a uniform grid, the model is intentionally non-linear. Regions directly in front of the agent are represented with high spatial resolution, reflecting the greater importance of information in the forward direction. Areas behind the agent or far above and below are represented much more coarsely. Each spatial cell contains transient Links connecting the cell to Atomic Thoughts representing recognized objects, sounds, or events. A "transient" Link is an ordinary link with persistence parameters set so that it will be quickly forgotten if not renewed. These Links may carry attributes such as estimated distance from the observer or motion and typically persist only briefly. In this way the spatial model captures the current state of the environment while the long-term knowledge about those objects remains stored in the knowledge network.

Because the spatial model uses the same Atomic Thought representation as the knowledge network, information can flow naturally between the two. Recognized objects activate related knowledge from the network so the addition of Fido to the spatial model requires the addition of a single Link.

The spatial model therefore serves as the interface between perception and reasoning. Incoming sensory information updates the model, knowledge retrieved from the network enriches it, and predicted events can be tested within it before actions are taken.

The following sections describe how the system selects which elements of this model to process in detail and how predicted outcomes are evaluated in order to g**uide behavior.**

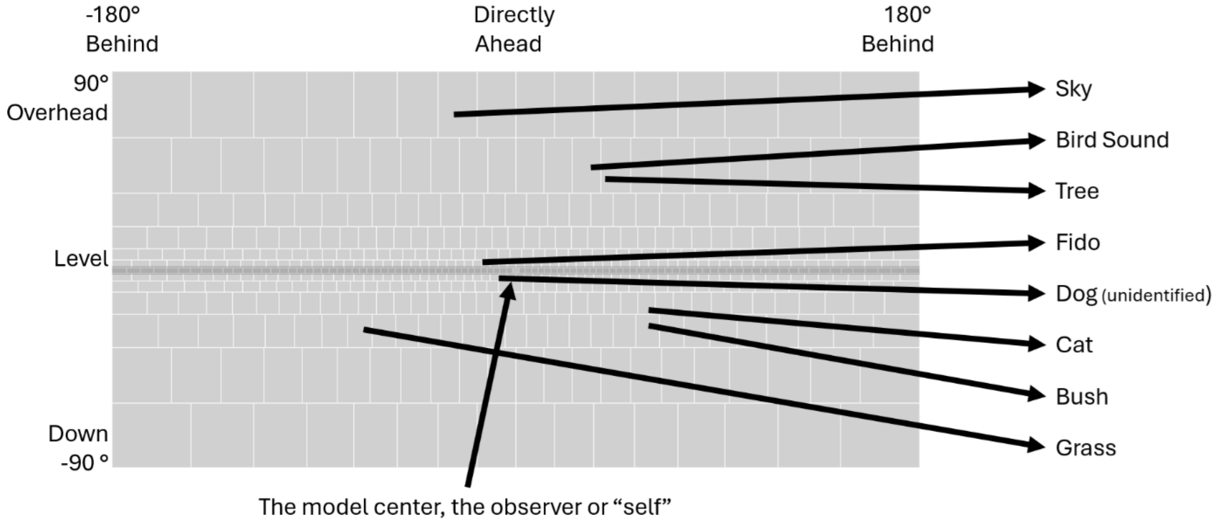


Figure 8. Egocentric Spatial Model. The spatial model represents the agent's current environment using an egocentric coordinate system centered on the observer. The horizontal axis spans $[-180^\circ, 180^\circ]$and the vertical axis $[-90^\circ, 90^\circ]$. Spatial resolution is non-linear: regions directly in front of the agent are represented with higher resolution, while areas behind or far above and below are represented more coarsely. Each spatial cell contains transient Links to Atomic Thoughts in the knowledge network representing recognized objects or events. These Links may include attributes such as estimated distance and motion. Because these Links are short-lived, the spatial model reflects only the current situation rather than long-term knowledge.

3.2.1 Why Reasoning Occurs in the Spatial Model

The knowledge network stores general knowledge accumulated through experience. It contains concepts, Links, attributes, and sequences describing many possible objects and events.

However, reasoning about the current situation cannot occur directly within this global structure. At any moment the network may contain millions of Thoughts describing situations that are unrelated to what the system is currently observing.

Instead, reasoning must occur within a representation that contains only the elements relevant to the present moment.

The spatial model serves this purpose. It acts as a temporary workspace representing the objects, events, and Links that currently exist in the system's environment. Concepts recognized through perception are inserted into this model, along with their spatial Links.

Once a situation is represented in the spatial model, the knowledge network can be consulted to retrieve additional attributes. In this way, the spatial model can be perceived as providing detail at a much higher level than the model itself actually represents.

The knowledge network therefore functions as long-term memory, while the spatial model represents the system's current understanding of the world. Reasoning operates within the spatial model because it contains the specific configuration of objects and events that are relevant at that moment.

3.3 Attention

The spatial model provides a structured representation of the system's current environment, but it also introduces a practical challenge. At any given moment the model may contain many objects, sounds, movements, and other sensory events. Processing all of these simultaneously in detail is not feasible. A mechanism is therefore required to determine which elements of the current situation deserve further analysis.

This function is performed by the attention system.

Sensory input arrives continuously from multiple modalities, including vision, sound, and other perceptual channels. These signals can be processed in parallel by specialized recognition systems, allowing many candidate objects and events to be identified simultaneously. As a result, the spatial model may quickly fill with recognized concepts and partially interpreted observations.

However, most higher-level cognitive operations cannot operate on many items at once. Pattern recognition processes—whether biological or artificial—generally compare an input pattern against a set of stored representations. Such comparisons are inherently sequential: a recognition system can evaluate only one candidate pattern at a time. To recognize multiple objects simultaneously would require multiple independent copies of the recognition mechanism.

Because of this limitation, the system must select which elements of the spatial model will be examined more carefully. The attention system performs this selection by maintaining a prioritized set of candidate items drawn from the spatial model.

Each candidate is evaluated using several signals that estimate its potential importance. These signals may include factors such as:

- Novelty — whether the item has been encountered recently
- Surprise — whether the observation contradicts expectations
- Motion — moving objects may indicate changing conditions
- Proximity — nearby objects may require immediate response
- Goal relevance — whether the item relates to current objectives
- Poor recognition — areas of unknown often attract additional scrutiny

Items that receive higher attention scores are selected for further processing. Once selected, the system can perform deeper analysis by consulting the knowledge network, retrieving additional attributes, or initiating sequence-based predictions about possible future events.

Attention therefore acts as the gatekeeper between perception and higher-level cognition. While sensory processing may generate many candidate interpretations in parallel, attention determines which of those interpretations will be examined in detail and used to guide reasoning and action.

The process is continuous. As the environment changes and new information arrives, the attention system continually reevaluates the items present in the spatial model, ensuring that the system's limited processing resources are directed toward the most relevant aspects of the current situation.

The following section describes how the system uses the selected elements of the spatial model to generate predictions about possible future states of the environment.

3.4 Simulation and Prediction

Once attention selects an element from the spatial model for deeper processing, the system attempts to predict how the current situation may evolve. This process is performed through simulation, in which the knowledge network is used to generate possible future states of the environment.

Prediction begins by examining sequences stored in the knowledge network. Experiences are recorded as chains of Links representing events that tend to occur in succession. When the system encounters the early portion of a previously observed sequence, it can search the network for Links that historically followed similar patterns.

For example, the network may contain a sequence such as:

[A → followed-by → B]
[B → followed-by → C]
[C → followed-by → D]

If the current spatial model corresponds to the situation:

A → B

then the system may predict:

C

as a likely next event.

However, prediction does not occur solely as an abstract lookup within the knowledge network. Instead, predicted events are inserted into the spatial model as hypothetical future states. In this way the system can simulate how the environment may change if current patterns continue.

For example, if the spatial model currently contains the object Fido moving toward a bush, previously learned sequences may suggest outcomes such as:

[Fido → enters → bush]
[Fido → emerges-from → bush]
[Fido → chases → cat]

Each of these potential outcomes can be projected into the spatial model as a candidate future state. These simulated states exist temporarily and do not modify the underlying knowledge network.

This process allows the system to explore multiple possible futures. Because each predicted state exists within the same spatial representation used for perception, it can interact with other elements of the environment and reveal additional consequences.

Prediction therefore functions as a form of internal experimentation. Rather than waiting for events to occur in the external world, the system can generate possible continuations of the current situation and examine their implications.

Importantly, this simulation process is guided by the same knowledge network that supports recognition and learning. Patterns extracted from past experience therefore influence not only how the system interprets the present, but also how it anticipates the future.

The predicted states generated during simulation are then evaluated according to the system's goals and internal measures of well-being. The next section describes how this evaluation process guides the selection of actions.

3.5 Evaluation and Well-Being

Prediction alone is not sufficient for intelligent behavior. An intelligent system must also determine whether a predicted outcome is desirable or harmful. The purpose of the evaluation system is to assess simulated future states and estimate their effect on the system's objectives.

In the Atomic Thought architecture this assessment is performed through an internal measure referred to as well-being. Well-being represents the system's current assessment of its overall condition and serves as the basis for evaluating potential actions. In its simplest implementation, well-being may be represented as a scalar value within a bounded range:

$\text{WellBeing} \in [-1, 1]$

Values near the positive end of this range represent favorable conditions, while negative values represent undesirable states.

The system learns Links between situations and their impact on well-being through experience. When events occur that improve or degrade the system's condition, Links can be formed connecting the relevant situation to the resulting change in well-being. Over time the knowledge network therefore accumulates information about which events tend to produce beneficial or harmful outcomes.

When the prediction system generates possible future states, each simulated state can be evaluated by examining the expected effect on well-being. This evaluation may consider multiple factors present in the spatial model, including objects, events, and predicted sequences.

For example, the presence of certain objects may historically correlate with favorable outcomes:

[food → improves → well-being]

while other events may correlate with negative outcomes:

[collision → decreases → well-being]

When such conditions appear within a simulated future state, the evaluation system estimates the resulting change in well-being.

Because prediction can generate multiple possible futures, the evaluation process may assign different well-being outcomes to each candidate state. These estimated outcomes provide the basis for selecting which course of action the system should pursue.

Importantly, well-being is not limited to physical survival. In more advanced systems the well-being measure may incorporate many different factors, including goal achievement, curiosity, learning opportunities, or social interactions. In principle the scalar value could also be extended to a vector representing multiple dimensions of evaluation.

The evaluation system therefore converts predicted future states into quantitative estimates of desirability. These estimates guide the system's decision process by identifying which potential outcomes are most favorable.

The following section describes how the system selects and executes actions that are expected to produce the most beneficial predicted outcomes.

3.6 Decision and Action

Once possible future states have been generated and evaluated, the system must determine how to act. The purpose of the decision system is to select actions that are expected to produce the most favorable outcomes according to the system's evaluation of well-being.

The decision process operates on the predicted future states produced during simulation. Each simulated state represents a possible continuation of the current situation if a particular action were taken or if current events were allowed to proceed. By estimating the effect of each possible state on well-being, the system can compare alternative courses of action.

Conceptually, the decision process can be viewed as selecting the action whose predicted consequences produce the most favorable change in well-being.

- Choose actions that maximize predicted well-being

For example, if the spatial model contains an action that has previously resulted in praise or positive feedback, the knowledge network may contain relationships indicating that such outcomes improve well-being. Simulation may therefore produce a future state in which the system performs that action and receives positive feedback. Because this predicted state results in an increase in well-being, the decision system may select the actions required to produce that outcome.

In other situations, predicted states may indicate harmful consequences. For example, if a rapidly approaching object is detected, predicted sequences may indicate a potential collision that would decrease well-being. In such cases the system may select actions that increase distance from the threat.

The chosen action is then transmitted to the system's motor or output mechanisms. These mechanisms may include physical movement, communication, or other interactions with the environment.

Once an action is executed, the environment changes and new sensory input arrives. This updated information enters the spatial model and the entire cognitive cycle begins again. In this way perception, prediction, evaluation, and action form a continuous loop in which the system constantly interprets the environment and adjusts its behavior accordingly.

Importantly, actions themselves also become part of the system's experience. The outcomes of actions are captured by the knowledge network as sequences linking actions to their consequences. Over time the system therefore learns which behaviors tend to improve or degrade well-being within particular contexts.

Through repeated interaction with the environment, the system gradually develops increasingly effective strategies for navigating the situations it encounters.

The following section summarizes how these components interact to form the complete cognitive architecture described in this paper.

3.7 The Cognitive Loop

The components described in the previous sections operate together as a continuous cognitive cycle through which the system interprets its environment and selects appropriate actions. Rather than functioning as isolated subsystems, perception, attention, prediction, evaluation, and action interact through the shared structures of the spatial model and the knowledge network as shown in Figure 9.

The process begins with sensory input. Signals from the environment are processed by specialized recognition systems that identify candidate concepts and events. These interpretations are inserted into the spatial model as transient Links connecting spatial cells to Atomic Thoughts stored in the knowledge network.

Because perception can generate many simultaneous observations, the spatial model may quickly contain numerous objects and events. The attention system selects among these candidates, prioritizing those that are most likely to be relevant according to factors such as novelty, motion, proximity, or goal relevance.

Once an element has been selected for deeper processing, the system consults the knowledge network to retrieve related information and previously observed sequences. This information allows the system to generate predictions about how the current situation may evolve.

Predicted events are inserted into the spatial model as hypothetical future states. By projecting these possibilities into the same representation used for perception, the system can simulate multiple potential outcomes of the current situation.

Each predicted state is then assessed by the evaluation system, which estimates the expected effect of that state on the system's well-being. These evaluations provide a quantitative basis for comparing alternative futures.

The decision system selects actions that are predicted to produce the most favorable outcomes. The chosen action is executed through the system's motor or output mechanisms, altering the environment.

These changes generate new sensory input, which again updates the spatial model and begins the cycle anew.

The overall cognitive loop can therefore be summarized as [10]:

Perception → spatial model → attention → prediction → evaluation → decision → action → environment update

The cognitive cycle described here may appear familiar and is not unique to this architecture. Variations of this perception–decision–action loop appear in many models of cognition and artificial intelligence, including perception–action cycles, sense–plan–act architectures, and other cognitive control frameworks.

Most intelligent systems must ultimately perform similar steps: perceive their environment, interpret the situation, predict possible outcomes, evaluate alternatives, and select actions.

What distinguishes the Atomic Thought architecture is not the existence of these phases but the uniform knowledge representation that connects them. In many architectures, perception, reasoning, prediction, and decision-making rely on different internal representations or specialized subsystems. As a result, information must often be translated between different structures as it moves through the system.

In the Atomic Thought architecture, all phases operate on the same underlying structure: a persistent network of Atomic Thoughts connected by Links. Concepts, attributes, relationships, and sequences are represented using the same primitive element. The spatial model, predictions, and learned experiences all reference the same relational network.

Because the same representation is used throughout the system, information discovered during perception can immediately influence prediction, evaluation, and decision-making without requiring conversion between different representations. The architecture therefore integrates perception, memory, reasoning, and action through a single relational knowledge structure rather than through a collection of loosely connected subsystems.

Throughout this process, experience is continuously incorporated into the knowledge network. Newly observed Links are captured, competing interpretations are tested through use, stable patterns are consolidated, and rarely used Links are pruned.

In this way the system's knowledge network evolves with experience while the spatial model represents the system's current understanding of the environment. Together these structures allow the system to interpret the present, anticipate the future, and select actions that improve its expected outcomes.

The remaining sections examine how this architecture compares with existing approaches and discuss considerations for practical implementation.

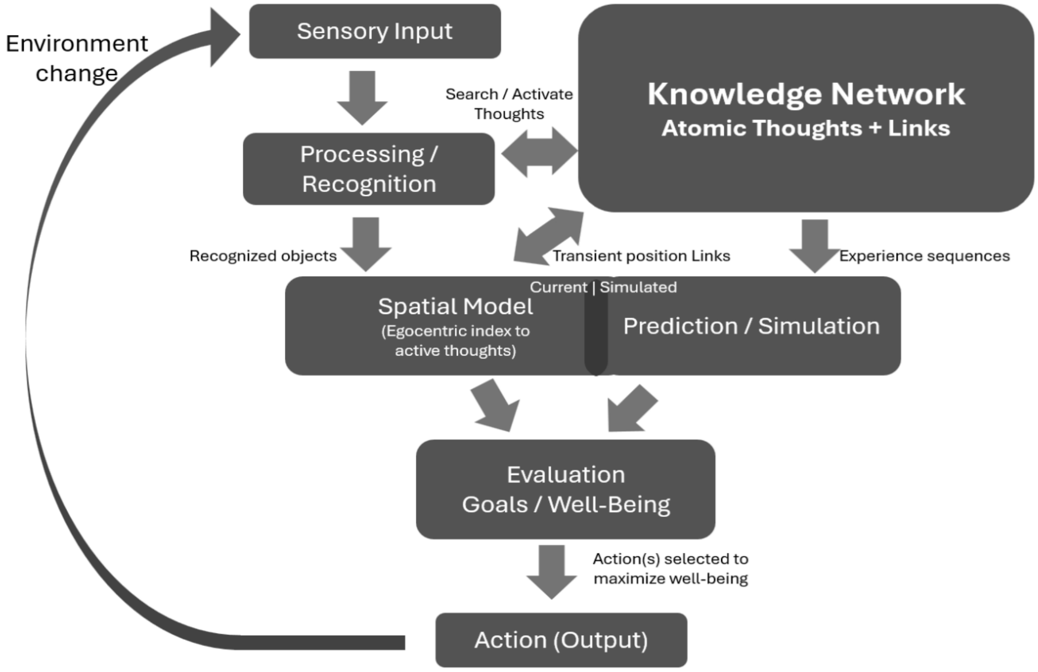


Figure 9. Atomic Thought Cognitive Architecture. Sensory input is processed to recognize objects and events, which are inserted into an egocentric spatial model that indexes active concepts within the knowledge network. The knowledge network stores persistent Atomic Thoughts and their relationships. Sequences of experience stored in the network are used to simulate possible future states within the spatial model. These predicted states are evaluated according to system goals and well-being, leading to action selection.

4 Conclusions and Future Work

4.1 Comparison with Existing AI Architectures

The Atomic Thought architecture differs from several widely used approaches in artificial intelligence. While many existing systems achieve impressive performance within specific domains, they typically rely on representations and learning mechanisms that differ substantially from the relational structure proposed here. Examining these differences helps clarify the motivation for the architecture described in this paper.

4.1.1 Transformer-Based Models

Transformer architectures have achieved remarkable success in language processing, image analysis, and many other applications. These systems operate by learning statistical links within large datasets and using those links to predict the continuation of sequences [7].

In transformer-based systems, knowledge is primarily encoded within large numerical parameter sets. During inference, an internal representation of the current input sequence is constructed through attention mechanisms operating over these learned parameters. Once the computation completes, however, this internal state is not preserved as a persistent representation of the world.

As a result, transformer models typically reconstruct context from the input provided during each interaction rather than maintaining a continuously evolving internal model. Concepts may appear within the generated output, but they do not exist as stable entities within a persistent knowledge structure that can accumulate experience across interactions.

The Atomic Thought architecture takes a different approach. Concepts and Links are represented explicitly as Atomic Thoughts within a persistent network. Rather than reconstructing context during each interaction, the system maintains a continuously evolving knowledge structure that grows through experience.

4.1.2 Knowledge Graphs

Knowledge graphs represent entities and relationships explicitly using nodes and edges. This structure allows facts about the world to be stored in a relational form and queried efficiently.

The Atomic Thought architecture shares some superficial similarities with knowledge graphs because both systems represent knowledge as networks of connected elements. However, knowledge graphs are typically static repositories designed for storing factual Links rather than supporting continuous learning and reasoning.

In conventional knowledge graphs, relationships are usually represented as edges connecting nodes. These edges typically do not possess the same expressive capabilities as nodes and cannot easily participate in higher-order relationships. There is usually a static, pre-defined set of allowable relationship types. Furthermore, the nodes themselves often contain a considerable amount of information. This means that searching information within nodes must be significantly different from searching the relationship information between nodes.

In contrast, the Atomic Thought architecture represents Links themselves as Atomic Thoughts. Because Links are first-class entities, they can carry attributes, participate in additional Links, and evolve through reinforcement and pruning. This allows the network to represent not only stored facts but also the processes through which knowledge is interpreted and revised.

4.1.3 Reinforcement Learning Systems

Reinforcement learning systems learn behaviors through trial-and-error interaction with an environment. By observing rewards and penalties associated with actions, these systems can gradually discover strategies that maximize long-term reward [6].

While reinforcement learning provides a powerful mechanism for learning behavior, the knowledge acquired by such systems is typically embedded within numerical value functions or policy networks. These representations often lack explicit conceptual structure and can be difficult to interpret or generalize beyond the environment in which they were trained.

The architecture proposed in this paper incorporates a concept similar to reward through the notion of well-being. However, rather than embedding learned behavior solely within numerical parameters, the system stores experiences as sequences and Links within the knowledge network. This structure allows experiences to be reused for recognition, prediction, and reasoning in addition to guiding action.

4.1.4 Architectural Perspective

Each of the approaches described above has demonstrated considerable success within its intended domain. Transformer models excel at large-scale pattern recognition, knowledge graphs provide structured representations of factual information, and reinforcement learning enables adaptive behavior in complex environments.

The Atomic Thought architecture does not attempt to replace these techniques directly. Instead, it proposes a different architectural foundation for representing and organizing knowledge within a cognitive system. By combining a persistent relational network with a spatial model of the current environment, the architecture attempts to unify perception, memory, prediction, and action within a single conceptual framework.

4.2 Biological Constraints and Plausibility

The Atomic Thought architecture is not only computationally feasible but also consistent with several known constraints of biological neural systems. While the exact mechanisms used by the brain remain an open area of research, a number of well-established properties of neurons strongly influence how biological cognitive systems must be organized. The design of the Atomic Thought architecture follows several of these constraints.

4.2.1 Neuronal Assemblies and Atomic Thoughts

An Atomic Thought represents the smallest unit of knowledge in the architecture. In a biological system, such a structure would not correspond to a single neuron but rather to a small assembly of neurons working together.

Individual neurons are unreliable and noisy, and meaningful cognitive representations are widely believed to emerge from coordinated activity among groups of neurons. Hebbian cell assemblies [7]and related concepts in neuroscience describe how groups of neurons can collectively represent a concept through their pattern of connectivity and activation.

The Atomic Thought representation is consistent with this idea. Rather than assuming that a concept or relationship corresponds to a single neuron, an Atomic Thought can be viewed as the coordinated activity of a small neural structure composed of many neurons. Such structures would provide redundancy and reliability, allowing concepts to persist despite the variability of individual neurons.

Cortical columns provide a plausible biological substrate for such structures [8]. Columns contain groups of neurons capable of representing related patterns of activity, and their repeated organization throughout the cortex suggests that they may serve as fundamental computational units of cognition.

4.2.2 Bidirectional Links

Human learning also suggests that Links are frequently accessible in both directions. When a person learns that Fido is a dog, they can immediately answer questions such as what is Fido? or name a dog. The relationship can be used in either direction without additional learning.

This behavior is difficult to explain using single synaptic connections, which are inherently directional: signals pass from presynaptic neurons to postsynaptic neurons. A single synapse therefore cannot directly represent a bidirectional relationship.

The ability to traverse Links in both directions suggests that relational knowledge must be implemented using assemblies of neurons connected through multiple pathways rather than through individual synapses. Within such an assembly, activation can propagate in several directions depending on which element of the relationship becomes active first.

The Atomic Thought representation naturally supports this property because Links are represented as structures capable of connecting three concepts simultaneously. In a biological implementation, such structures could be realized through assemblies of neurons connected by multiple reciprocal synaptic pathways.

4.2.3 Relational Structure Rather Than Numerical Precision

Many modern artificial intelligence systems depend heavily on floating-point numerical representations. Neural network activations, attention weights, and gradient updates rely on precise numerical values and continuous mathematical operations.

While such representations are efficient on digital computers, they are difficult to reconcile with biological neurons. Neurons communicate through discrete spikes and operate on time scales measured in milliseconds. Representing precise floating-point values through spike timing would require complex and inefficient mechanisms. Further, synapse weights, rather than representing specific values, are not stable and reliable enough to represent real numbers and therefore can only represent a small number discrete values.

The Atomic Thought architecture avoids this difficulty by representing meaning primarily through relational structure rather than numerical magnitude. Concepts and Links are expressed through the presence of Links and the propagation of activation across those Links. The Link weights are used competitively so the specific values are not critical. Meaning is carried primarily by the pattern of connections between concepts rather than by precise numerical values.

4.2.4 Speed Constraints and Direct Associations

Another important constraint of biological cognition is the speed of neural signaling. A typical synaptic transmission requires several milliseconds, and a chain of only a few synapses already introduces tens of milliseconds of delay.

Because of this limitation, cognitive processes must rely on very short chains of activation. Large search procedures or iterative numerical computations would be prohibitively slow in neural systems.

Artificial intelligence systems often perform explicit searches through large structures or rely on repeated matrix operations. While these operations are efficient on modern hardware, they would be impractical if implemented directly with biological neurons.

Instead, biological cognition appears to rely on direct associative links between related concepts. Recognition does not occur by searching a large database of possibilities but by activating concepts that are already directly connected to the observed components of the input.

For example, the recognition of a word may occur because individual letter concepts are directly linked to the word they compose. Similarly, notes within a melody may be directly linked to the higher-level musical pattern they form. When enough components become active, activation propagates through these existing links and the higher-level concept becomes active.

The Atomic Thought architecture adopts this same principle. Rather than searching through large sets of possibilities, the system relies on direct Links connecting related Thoughts. Activation propagates through these Links, allowing recognition and inference to occur through short chains of relational activation.

4.2.5 Sequences and Temporal Learning

Temporal Links between events are naturally represented as chains of associations in biological systems. Mechanisms such as spike-timing dependent plasticity (STDP) [9] reinforce synapses when one neuron consistently fires shortly before another.

This type of learning naturally produces sequential structures in which activation propagates from one event representation to the next. Such mechanisms provide a biologically plausible foundation for representing sequences of events.

The sequence Links used in the Atomic Thought architecture follow this same principle. Experiences are stored as chains of Links connecting successive events. When the early portion of a sequence becomes active, activation naturally propagates through these links, allowing the system to anticipate likely future events.

4.2.6 Summary

Taken together, these observations suggest that cognitive architectures compatible with biological systems should exhibit several properties:

- Concepts represented by assemblies of neurons rather than single cells
- Links accessible in multiple directions
- Meaning encoded through relational structure rather than numerical precision
- Recognition based on direct associative links rather than large searches
- Sequential knowledge represented through chains of learned associations

The Atomic Thought architecture satisfies each of these constraints. While the precise neural implementation of conceptual Links remains an open question, the mechanisms described here are consistent with known properties of neuronal networks and therefore represent a plausible framework for biologically inspired intelligence systems.

4.2.7 Implementation Considerations

The architecture described in this paper is intended as a conceptual framework for constructing cognitive systems based on persistent relational structures. While the preceding sections describe the principles underlying the architecture, practical implementation requires addressing several engineering considerations.

These considerations primarily involve the representation of Atomic Thoughts, the management of the knowledge network as it grows, and the efficiency of search operations performed during perception and reasoning.

4.2.8 Representation of Atomic Thoughts

In practice, each Atomic Thought must store a small amount of information describing its Links to other thoughts and its history of use. These Links form the Links that connect the network.

An implementation may store each Atomic Thought as a lightweight structure containing references to related thoughts along with a small set of attributes describing its activity. Such attributes may include the time the thought was created, the time it was last used, and the number of times it has participated in searches or predictions.

These measures support the learning processes described earlier. Frequently used Links can be reinforced through repeated activation, while rarely used Links gradually decay and may eventually be pruned from the network.

Because each Atomic Thought is small and structurally uniform, the network can scale to large numbers of elements while maintaining a consistent representation.

4.2.9 Network Growth and Pruning

As the system encounters new experiences, the number of Atomic Thoughts and Links within the network will grow. Without mechanisms for controlling this growth, the network could eventually become inefficient to search.

The capture–compete–consolidate–prune cycle described earlier provides a natural mechanism for managing network size. Newly observed Links are initially captured with minimal filtering. Over time, Links that are frequently used during recognition or prediction become reinforced and consolidated.

Links that are rarely used gradually lose persistence and may eventually be removed from the network. This pruning process ensures that the network increasingly reflects patterns that are actually useful for interpreting the environment.

4.2.10 Efficient Knowledge Search

Many operations within the architecture involve searching the network for related concepts or sequences. Efficient traversal of the network is therefore essential.

Because Links explicitly connect related Atomic Thoughts, searches can propagate through the network using graph traversal techniques. The traversal properties described earlier—such as transitivity and inheritability—determine how Links are followed during search.

In practice, searches are often guided by partial information obtained from perception or by concepts currently active within the spatial model. This allows the system to explore only the portions of the network that are relevant to the current situation rather than scanning the entire structure.

Prioritization mechanisms, such as those used by the attention system, can further limit the search space by focusing computation on the most promising candidates.

4.2.11 Integration with Perceptual Systems

The architecture assumes the existence of perceptual subsystems capable of identifying candidate concepts from sensory input. These subsystems may employ a variety of techniques, including neural networks or other pattern-recognition methods.

Once a perceptual subsystem identifies a concept, it simply activates the corresponding Atomic Thought within the knowledge network and links it into the spatial model. In this way the architecture can incorporate modern machine learning systems as perceptual components without requiring that all knowledge be encoded within neural parameters.

4.2.12 Scalability and Practical Systems

Although the architecture described here is conceptually simple, its implementation may involve networks containing large numbers of Atomic Thoughts and Links. Modern computing systems are capable of storing and traversing large graph structures efficiently, making such implementations feasible.

Moreover, the distributed nature of the network allows learning and reasoning processes to occur incrementally as new experiences are encountered. The system does not require retraining on large datasets; instead, knowledge accumulates continuously through interaction with the environment.

These characteristics make the architecture well suited for systems that must operate in dynamic environments where knowledge grows and evolves over time.

The following section discusses broader implications of this architecture and potential directions for future development.

4.3 Discussion and Future Directions

The architecture described in this paper proposes a different foundation for artificial intelligence systems. Rather than storing knowledge primarily within large numerical parameter sets, the Atomic Thought architecture represents knowledge as a persistent relational structure that evolves through experience. Concepts, Links, sequences, and predictions all emerge from the same primitive representation: Atomic Thoughts connected by Links.

One important implication of this design is that learning occurs continuously through interaction with the environment rather than through discrete training phases. New observations are incorporated directly into the knowledge network, and the structure of the network evolves through repeated use. Patterns that prove useful become reinforced, while rarely used Links gradually disappear. In this way the system's knowledge structure reflects its accumulated experience rather than a fixed dataset.

Another implication is that reasoning, prediction, and perception operate within a shared representational framework. Because the spatial model and the knowledge network use the same underlying representation, information discovered during recognition can immediately influence prediction and decision-making. This unified structure contrasts with many existing systems in which perception, reasoning, and planning rely on separate representations.

The architecture also provides a natural mechanism for explainability. Because knowledge is stored explicitly as Links between identifiable concepts, the system can trace how a particular interpretation or prediction was derived. Rather than relying on opaque numerical representations, the system's reasoning can be examined by following the Links connecting relevant Atomic Thoughts.

Several areas remain open for further research. Efficient algorithms for large-scale network traversal will be important as the number of Atomic Thoughts grows. Mechanisms for prioritizing searches and managing activation within the network will also play a key role in maintaining performance as the system encounters increasingly complex environments.

Another area of exploration involves the interaction between the relational knowledge network and modern machine learning techniques. Neural networks may provide powerful perceptual subsystems capable of recognizing patterns in sensory input, while the relational architecture described here could provide the persistent conceptual structure required for higher-level cognition.

Finally, the architecture raises questions about how complex cognitive abilities might emerge from relatively simple primitives. Capabilities such as abstraction, analogy, and planning may arise naturally as the knowledge network grows and the system accumulates experience. Investigating how these abilities develop within such a framework represents an important direction for future work.

4.4 Conclusion

This paper has presented an architectural framework for constructing intelligent systems based on persistent relational structures. At the core of the architecture is the concept of the Atomic Thought, a primitive unit representing either a concept or a relationship within a continuously evolving knowledge network.

By representing Links as Atomic Thoughts, the system allows Links themselves to possess attributes and participate in other Links. This uniform representation enables complex knowledge structures to emerge from simple relational components.

The architecture combines this persistent knowledge network with a spatial model that represents the system's current environment. Perception inserts recognized concepts into the spatial model, attention selects elements for deeper processing, prediction generates possible future states, evaluation estimates their effects on well-being, and the decision system selects actions expected to produce the most favorable outcomes.

Together these processes form a continuous cognitive loop through which the system interprets its environment and adapts its behavior through experience.

While many current artificial intelligence systems excel at pattern recognition and statistical prediction, they often lack a persistent internal structure capable of representing evolving knowledge about the world. The architecture described here explores an alternative approach in which understanding arises from a growing network of concepts and Links that persist across time.

Although significant work remains to develop practical implementations of this framework, the underlying principles suggest that intelligence may depend less on scaling numerical models and more on constructing architectures capable of representing and evolving structured knowledge.

References

[1] Newell, A. (1990). Unified Theories of Cognition. Harvard University Press.

[2] Laird, J. (2012). The Soar Cognitive Architecture. MIT Press.

[3] Sowa, J. (1984). Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley.

[4] Hogan, A., Blomqvist, E., Cochez, M., et al. (2021). Knowledge Graphs. ACM Computing Surveys.

[5] Vaswani, A., et al. (2017). Attention Is All You Need. NeurIPS.

[6] Sutton, R., & Barto, A. (2018). Reinforcement Learning: An Introduction.

[7] Hebb, D. (1949). The Organization of Behavior.

[8] Mountcastle, V. (1997). The columnar organization of the neocortex.

[9] Bi, G., & Poo, M. (1998). Synaptic modifications in cultured hippocampal neurons dependent on spike timing.

[10] Fuster, J. (2015). The Prefrontal Cortex.

[11] Simon, C. Brain Simulator Thought: A Research Platform for Brain-Inspired AI. Available at: https://github.com/FutureAIGuru/BrainSim-Thought