

A New Information Architecture: a Synthesis of Structure, Flow, and Dialectic

Rico A. R. Picone^{1,2} and Bryan Powell¹

¹ Dialectica, LLC

² Department of Mechanical Engineering, Saint Martin's University

Abstract. We introduce a new information architecture that is a synthesis of an information structure, information flow (e.g. narrative), and information dialectic (its evolution). Insights from the memory mnemonic the *method of loci* and the philosophical process of *dialectic* are introduced. Three cumulative syntheses are presented: the organic architecture with (1) the hierarchical architecture with (2) the sequential architecture with (3) the dialectical method.

1 Background

Since the conception of personal computing, it has been the goal of various researchers and designers to create a computing system that can be a medium for human thought [3]. Certainly, we have come a long way toward realizing this; but we remain encumbered by the standard information architectures. We will examine each of the common information architectures and find their strengths and weaknesses, before synthesizing them into a new information architecture.

1.1 Information Architecture

Information architecture [1, p. 88] is the manner in which pieces of information are interrelated. The basic unit of an information architecture is the *atom* (e.g. file, paragraph, etc.). Atoms are logically related via *categories* (e.g. directory, tag, etc.). Categories can be considered as *sets* of atoms.³

Hierarchical Architecture. The *hierarchical architecture* is an architecture in which atoms and categories are organized with parent-child relationships in a *tree*-structure. The categories are arranged to encompass one-another, as in the case of a computer file-system. Then atoms populate the structure at various levels (see Fig. 1).

The advantages of this architecture include that it gives an intuitive structure to the information (with its spatial analogies like “up” and “in”), that it can be

³ It should be noted that this definition of a category is not the same as that of *category theory*. Our definition of a category is in the classical sense of a quality or attribute.

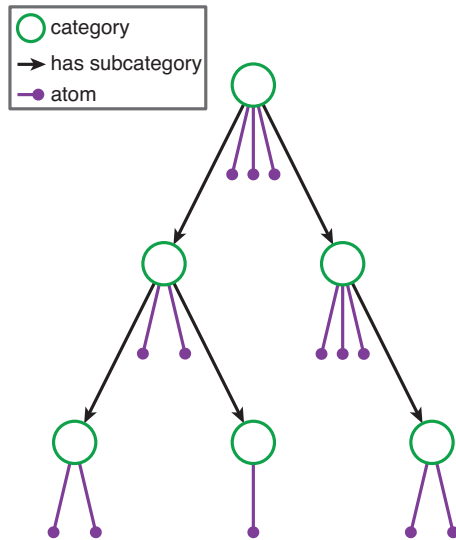


Fig. 1: A graph of the hierarchical architecture in terms of parent-child relations.

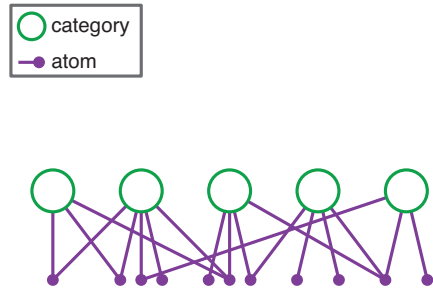


Fig. 2: A graph of the organic architecture. Its “flat” structure allows all categorizations.

easily instantiated in a computing device, and that — in many instantiations — it can be navigated with minimal atoms and categories displayed at each level.

The hierarchical architecture has limitations, however. It cannot allow certain categorizations of atoms. For instance, if an atom is associated with categories that lie in different “branches” of the tree, this cannot be properly expressed. Another limitation is that there is only a single path to each atom. We will address the concept of path in Sec. 3.3, but for now we can think of it as being like a file-path in a standard computing system. With only a single path to it, a browser of the architecture must know all the “right” moves in order to arrive at the atom.

Organic Architecture. The *organic architecture* is one which allows atoms to be categorized with any combination of categories, but does not directly relate the categories to each other. This architecture is instantiated in “tagging” systems, for instance, in which a tag can be considered a category.

The lack of explicitly defined relations among the categories gives this architecture its characteristic “flat” structure (see Fig. 2). It is a significant improvement (over the hierarchical architecture) in connectivity, allowing all possible relationships to be expressed, but the structure among the categories is opaque, and, in fact, undefined.

This is problematic when the architecture is instantiated into a computing system that is to be navigated (i.e. browsed). Where does one begin to navigate? What should be displayed at a given state of navigation? In Sect. 2.1, we will continue to discuss these issues.

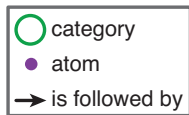


Fig. 3: a graph of the sequential architecture.

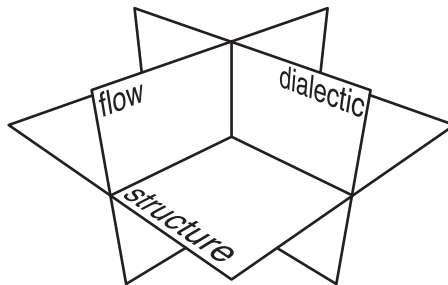


Fig. 4: three planes of information architecture.

Sequential Architecture. The *sequential architecture* is one which arranges atoms or categories in sequences with the relation “is followed by” (see Fig. 3). This architecture is that of narrative, film, and music—and is generally the architecture of the human experience of events in time. For this reason it is ubiquitous, although it is very limited, structurally.

It is common to create a synthesis of the sequential and hierarchical architectures, as in a document, such as this, with sections. Nominally, each subsection is related only to its (parent) section, although this is rarely actual. This synthesis is obviously fruitful, but inherits the limitations of the hierarchical architecture. We use this as an example of a successful synthesis of information architectures. What follows is a series of three cumulative syntheses that sublates the standard architectures in an attempt to capture the structure of human thinking. The primary objective of this work is to enable a computer—instantiating an information architecture—to be used as a better medium for human thinking.

2 Structure

We will distinguish among three planes of information architecture: the *structure plane*, the *flow plane*, and the *dialectic plane* (see Fig. 4).

Definition 1 (structure plane). The structure plane of an information architecture is the set of logical relations among the categories and atoms. \square

The structure plane is often the focus of an information architecture, but it can take different forms. The hierarchical architecture has structure in its category-category and atom-category relations, whereas the organic architecture has structure only in its atom-category relations.

2.1 Synthesis I: Organic and Hierarchical Architectures

The first synthesis is of the organic and hierarchical architectures. We begin with the cornerstone of the organic hierarchy: *each atom may be associated with any categories*. Beginning here ensures that the architecture will preserve the capacity for completely general categorization of information.

However, category-category relations are undefined. This is a limitation, especially for navigation and discovery.⁴ The synthesis of the organic and hierarchical architectures hinges on this precept: *a hierarchical architecture can be defined as implicit in an organic architecture*.

We call this synthesis an *organic hierarchy*. It stands alone as an information architecture, but two more cumulative syntheses will include it in the following sections. The organic hierarchy is described in the following series of mathematical definitions. The definitions need not be a hurdle to the understanding of the architecture. On the first reading, it is sufficient to skip them and proceed to Sect. 3.

Definition 2 (subcategory). Given a collection of categories \mathcal{C} , let $X \in \mathcal{C}$ be a nonempty proper subset of $Y \in \mathcal{C}$. We call X a *subcategory* of Y . \square

Definition 3 (graph nodes). Let \mathcal{C} be a collection of categories; let \mathcal{U} be the union of all categories in \mathcal{C} ; and let \mathcal{I} be the set of all possible nonempty recursive intersections of the elements of \mathcal{C} . Let *graph nodes* be defined as

$$\mathcal{N} = \mathcal{U} \cup \mathcal{C} \cup \mathcal{I}. \quad \square \quad (1)$$

Definition 4 (graph level function). The *graph level function* $L : \mathcal{N} \rightarrow \mathbb{Z}$ is defined to be equal to 0 if the node $n \in \mathcal{U}$, equal to 1 if the node $n \in \mathcal{C}$, and equal to the number of categories $C \in \mathcal{C}$ for which the node n was the intersection, otherwise. Therefore, $0 < L(n) \leq \text{card}(\mathcal{C})$ for $n \in \mathcal{N}$. Let \mathcal{L}_i be the collection of sets for which $L(n) = i$ for $n \in \mathcal{N}$, and let \mathcal{L}_i be indexed such that we can denote a relation \mathbf{r}_i^{jk} to be that from element j of \mathcal{L}_i to element k of \mathcal{L}_{i+1} .⁵ \square

Definition 5 (metacategories). Given a set of nodes $N \subseteq \mathcal{L}_i$, we can construct a collection \mathcal{M} of all recursive unions of the elements of N . We call the elements of \mathcal{M} *metacategories* of N .⁶ Furthermore, let the map **meta** be defined as $\mathbf{meta}(N) = \mathcal{M}$. Finally, let \mathcal{M}^d be the set containing the elements of **meta**(N) that were constructed from the unions of d nodes; we call d the *metacategory depth*. \square

Example 1 (nodes, levels, and metacategories). Given a collection of categories $\mathcal{C} = \{X, Y, Z\}$, the union of all categories is $\mathcal{U} = \{X \cup Y \cup Z\}$. The set of all

⁴ Discovery is the activity of learning something new about an information system.

⁵ In order to minimize notation, at times we refer to a node by its index.

⁶ All original nodes in N are also metacategories because the empty set \emptyset is a subset of every set, and the union of any set with \emptyset is just that set. Given ν nodes, the total number of metacategories is $\sum_i^\nu \binom{\nu}{i}$.

recursive intersections of the categories is

$$\mathcal{I} = \{X, Y, Z, X \cap Y, Y \cap Z, X \cap Z, X \cap Y \cap Z\}. \quad (2)$$

Therefore, assuming all intersections are nonempty, the set of nodes is

$$\mathcal{N} = \{X \cup Y \cup Z, X, Y, Z, X \cap Y, Y \cap Z, X \cap Z, X \cap Y \cap Z\}. \quad (3)$$

The collections of graph nodes at each level are

$$\mathcal{L}_0 = \{X \cup Y \cup Z\}, \quad (4a)$$

$$\mathcal{L}_1 = \{X, Y, Z\}, \quad (4b)$$

$$\mathcal{L}_2 = \{X \cap Y, Y \cap Z, X \cap Z\}, \text{ and} \quad (4c)$$

$$\mathcal{L}_3 = \{X \cap Y \cap Z\}. \quad (4d)$$

If we give **meta** the set of nodes $N = \{X, Y, Z\} \subseteq \mathcal{L}_1$, we get the collections of metacategories of N at each depth:

$$\mathcal{M}^1 = \{X, Y, Z\}, \quad (5a)$$

$$\mathcal{M}^2 = \{X \cup Y, X \cup Z, Y \cup Z\}, \text{ and} \quad (5b)$$

$$\mathcal{M}^3 = \{X \cup Y \cup Z\}. \quad \square \quad (5c)$$

We define three types of *graph relation* (or graph edge).

Definition 6 (s-relation). The relation **s** is *has a priori subcategory*. An *a priori* subcategory is a subcategory that is required by construction, and is independent of the contents of the categories. For instance, $A \cap B$ is an *a priori* subcategory of A , assuming it is not the empty set. An **s**-relation $\mathbf{s}_i^{j,k}$ is defined from tail-node j of \mathcal{L}_i to head-node k of \mathcal{L}_{i+1} if node k is an *a priori* subcategory of j . \square

Definition 7 (vs-relation). Consider a node j of \mathcal{L}_i . Let N be the set of (**s**-relation) head-nodes in \mathcal{L}_{i+1} for which j is the common tail-node. Let \mathcal{M}_{ij} be the set of metacategories of N , $\mathcal{M}_{ij} = \mathbf{meta}(N)$. A *has visible subcategory* **vs**-relation with tail-node j is an **s**-relation with head-node an element of the set of nodes used to construct \mathcal{M}_{ij} that have the minimum depth d_{\min} and that equal j' : the tail-node j intersected with the union of the elements of N . \square

Definition 8 (hs-relation). An **s**-relation that is not a **vs**-relation is defined as a *has hidden subcategory* **hs**-relation. \square

Definition 9 (visible atom). For a node $j \in \mathcal{L}_i$, a *visible atom* is an atom that is in the set $j \setminus j'$, where j' is the (**s**-relation) tail-node j intersected with the union of all head-nodes with tail-node j . \square

Definition 10 (graph of categories). Together, the set of nodes \mathcal{N} and **hs**- and **vs**-relations define a *graph of categories*. The graph can be *navigated* by

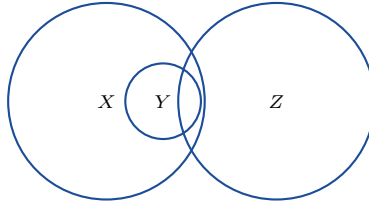


Fig. 5: Venn diagram for the categories of Example 2.

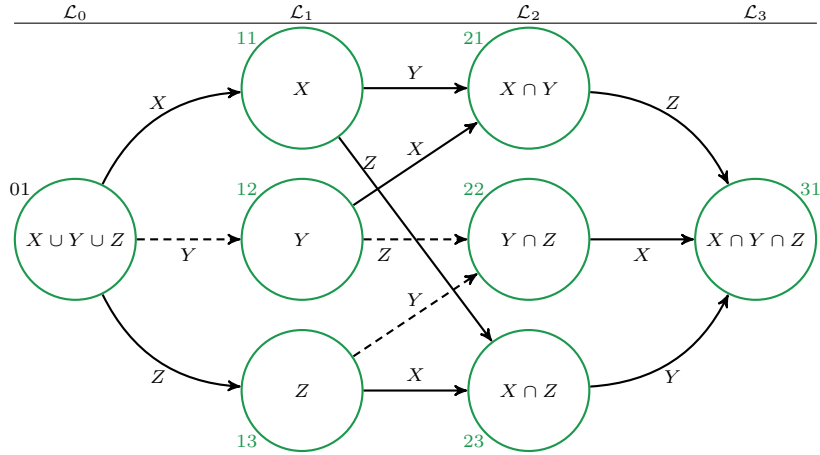


Fig. 6: graph of categories for Example 2. Dashed lines are **hs**-relations and solid lines are **vs**-relations.

beginning at any node and following an edge to an adjacent level (\mathcal{L}_i to \mathcal{L}_{i+1} or to \mathcal{L}_{i-1}). When the level increases, the edge represents the intersection of the current node with a single category, and can therefore be represented as “selecting” an additional category. When the level decreases, the edge represents the union of the current node with a single category, and can therefore be represented as “deselecting” that category. At a given node, visible atoms and **vs**-relations should be displayed; **hs**-relations should be somehow accessible, but not displayed. \square

Example 2 (graph of categories). Given a collection of categories $\mathcal{C} = \{X, Y, Z\}$, and the set relations defined by the Venn diagram Fig. 5, we can construct the graph shown in Fig. 6 using Def. 10 as follows. The graph nodes were already discussed in Example 1.

The *has subcategory* graph relations **s** are determined by construction, as long as they are nonempty. The *has visible subcategory* **vs**-relations for a given node can be determined by examining the **s**-relations with the node as its tail. The heads of these **s**-relations can be constructed into a set of metacategories, from which we can determine visibility. See Table 1, which lists the metacategories for

each node. Let's examine Node 01 ($X \cup Y \cup Z$), for instance. It is the tail for three **s**-relations because X , Y , and Z are all subcategories. Then the metacategories, divided into sets of depth 1–3, are

$$\mathcal{M}_{01}^1 = \{X, Y, Z\}, \quad (6a)$$

$$\mathcal{M}_{01}^2 = \{X \cup Y, Y \cup Z, X \cup Z\}, \text{ and} \quad (6b)$$

$$\mathcal{M}_{01}^3 = \{X \cup Y \cup Z\}. \quad (6c)$$

Table 1: metacategories used to determine **hs**- and **vs**-relations for Example 2. Nodes are referred to by Level i and index j .

Node ij	\mathcal{M}_{ij}^1	\mathcal{M}_{ij}^2	\mathcal{M}_{ij}^3
01	$\{X, Y, Z\}$	$\{X \cup Y, Y \cup Z, X \cup Z\}$	$\{X \cup Y \cup Z\}$
11	$\{X \cap Y, X \cap Z\}$	$\{X \cap (Y \cup Z)\}$	–
12	$\{X \cap Y, Y \cap Z\}$	$\{Y \cap (X \cup Z)\}$	–
13	$\{X \cap Z, Y \cap Z\}$	$\{Z \cap (X \cup Y)\}$	–
21, 22, 23	$\{X \cap Y \cap Z\}$	–	–

Let us consider those metacategories of depth $d = 1$. Evaluating the metacategories shows that no single category equals $j' = (X \cup Y \cup Z) \cap (X \cup Y \cup Z) = X \cup Y \cup Z$. Now consider those metacategories of depth $d = 2$. Only $X \cup Z = j'$; therefore, the **s**-relations from Node 01 to 11 (X) and 13 (Z) are **vs**-relations. The only visible atoms at \mathcal{L}_0 are those that are uncategorized. Similarly, for Node 11 (X), which is at Level 1,

$$\mathcal{M}_{11}^1 = \{X \cap Y, X \cap Z\} \text{ and} \quad (7a)$$

$$\mathcal{M}_{11}^2 = \{(X \cap Y) \cup (X \cap Z)\} = \{X \cap (Y \cup Z)\}. \quad (7b)$$

Let us consider those metacategories of depth $d = 1$. Evaluating the metacategories shows that no single category equals $j' = X \cap ((X \cap Y) \cup (X \cap Z))$. Considering the single metacategory at the maximum depth $d = 2$, we can see that, by definition, it evaluates to j' , and it requires all metacategories for construction. Therefore, all **s**-relations with tail-node X are **vs**-relations. This is why we gray-out the maximum-depth metacategories in the table; they needn't be evaluated in order to know the result. The visible atoms at this node are $X \setminus (Y \cup Z)$.

Proceeding in a similar manner for each node, we define all **s**-relations as either **vs**- or **hs**-relations, and define the visible atoms for each node. \square

3 Flow

The *method of loci* is an ancient memory mnemonic still in use by most memory champions [2]. The technique requires a subject to “place” memories in an

imaginary world familiar to the subject. It turns out that the technique is not then used as random-access memory in which the user can access any memory at will; rather, each memory-location is accessed *in series* by “journeying” through the imaginary world [2, p. 93].

This architecture is already expressed in the standard sequential information architecture. This architecture is fundamentally different than the other information architectures mentioned, which express the logical relations among sequences and atoms (i.e. they are *structural*). The sequential architecture has no structure itself; rather, *it expresses the movement of thinking through information structure*. It moves in another dimension, which we call the *flow plane* of information architecture.

Definition 11 (flow plane). The flow plane of an information architecture is the sequential presentation of atoms or categories. \square

3.1 Flow and Structure in Standard Information Architectures

As in the method of loci, thinking flows from one idea to the next, not necessarily in some order prescribed by the structure, but through the structure, which provides context for each idea. This is what separates the concept of flow from that of *path*. A path is the logical categorization of idea, whereas flow is a sequential presentation of ideas.

The quintessential example of a flow is a narrative, which is often presented in terms of chapters, sections, and subsections. This is a synthesis of the sequential and hierarchical architectures. Its utility is undeniable; it is the structure of nearly all scientific and technical literature. However, it inherits the limitations of the hierarchical architecture in that a flow cannot move freely through the structure due to its restrictions on logical categorizations.

3.2 Synthesis II: Flow and Structure

The second synthesis is the sublation of the organic hierarchy proposed in Sect. 2.1 and the sequential architecture. Therefore, it is the synthesis of structure and flow. This synthesis revolves around two insights, one presented in each of the following examples.

Example 3 (flow not limited by structure). The first insight is that flows should not be limited by structure. Consider the simple example of a chapter that compares the works of Søren Kierkegaard and Friedrich Nietzsche. In a hierarchical architecture, the dilemma is whether this belongs in **Kierkegaard/Nietzsche** or **Nietzsche/Kierkegaard**. It imposes a structure on the flow that does not exist.

In an organic hierarchy, however, a flow has no such limitations. The cornerstone of the organic hierarchy is that each atom — the **Kierkegaard-Nietzsche** chapter, in this case — may be associated with any category. Therefore, in any

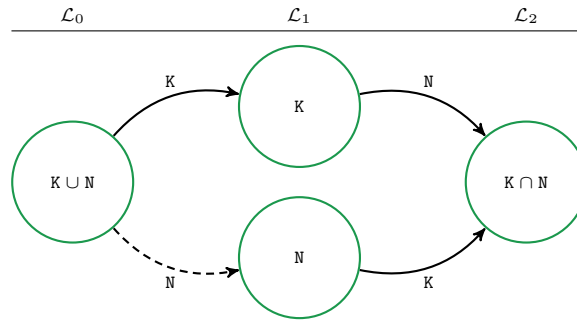


Fig. 7: graph of categories for Example 3. Dashed lines are **hs**-relations and solid lines are **vs**-relations. **K** is the category **Kierkegaard** and **N** is the category **Nietzsche**.

instantiation of this architecture, it will be available to a user under *both Kierkegaard/Nietzsche* and *Nietzsche/Kierkegaard*. And yet, the structure among the categories is not lost, as in a purely organic architecture. It still may be the case that in a given information system every atom associated with **Nietzsche** is also associated with **Kierkegaard**, but some atoms associated with **Kierkegaard** are not associated with **Nietzsche**. Therefore $\text{Nietzsche} \subset \text{Kierkegaard}$, and **Nietzsche** would be hidden within **Kierkegaard** in graph navigation, as illustrated in Fig. 7.

This would look identical to a user browsing the structure as a hierarchy with **Nietzsche** a subcategory of **Kierkegaard**, *but it does not restrict future categorizations*, unlike the hierarchy. Moreover, unlike in a hierarchy, there is nothing prohibiting a user from navigating to **Nietzsche** immediately (by manual input). It is an allowed, but not a visible (displayed to the user), navigation. \square

Example 4 (travel journal). The second insight that defines the synthesis of flow and structure is that the structuring of a flow must be highly granular. Categorizing a flow or even, say, a chapter, as a single atom is typically not sufficiently granular to capture the movement of the flow through the structure. Consider a travel journal as a flow with contents structured in an organic hierarchy, as illustrated in Fig. 8.

We will consider the first five paragraphs of the journal in question, *Journal 12*. Let all paragraphs be atoms associated with the categories **Journal**, **Travel**, and **Italy**. Furthermore, let paragraphs 1, 2, and 3 be associated with **Mosaics**; paragraphs 1–3 be associated with **Mausoleums**; paragraphs 3 and 4 associated with **Justinian I**; paragraphs 3–5 associated with **Byzantine Empire**; and paragraphs 4 and 5 associated with **Byzantine Artwork**. The flow side of the figure shows these categorizations.

If we assume that this journal lies in a broader structure of categories and atoms, we can construct a possible structural view, as shown at right. The graph has been navigated along one of many possible paths. (The order of the categories selected is irrelevant to a given node, so the user may have come along many

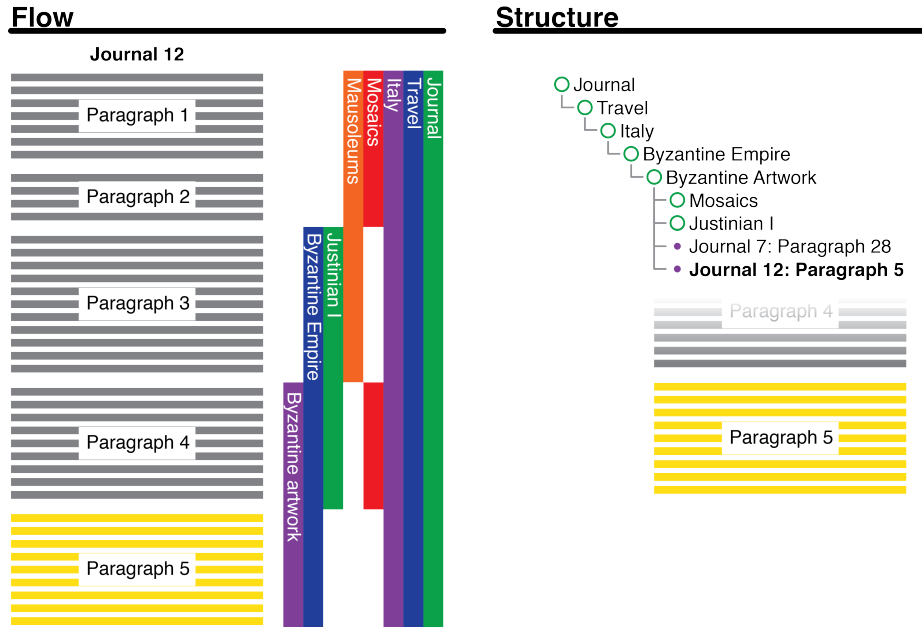


Fig. 8: illustration of a travel journal for Example 4. *Paragraph 5* is shown in both a flow (*Journal 12*) and a structure.

paths to that node.) The node shown appears to be simply **Byzantine Artwork**, but it is, in fact

$$\text{Journal} \cap \text{Travel} \cap \text{Italy} \cap \text{Byzantine Empire} \cap \text{Byzantine Artwork}. \quad (8)$$

The intersection operator commutes, so any order is equivalent, and therefore any path including the same categories is as well. The further intersection with the categories **Mosaics** and **Justinian I** is possible, which would reveal the atom paragraph 4. At the node shown, however, there are two visible atoms: *Journal 7: Paragraph 28* and *Journal 12: Paragraph 5*. *Paragraph 5* is associated with all the categories in the path, but no others. Therefore it is visible, and can (depending on the instantiation) be previewed in context, as shown.

This shows the possibility of discovery in this architecture. Let us imagine that the journaler reviews the journal later and is struck by something about *Paragraph 5*. Now interested to know if any related information in her broader information system, she views the structure and discovers that *Journal 7: Paragraph 28* was also associated with the same categories, and that other information that is related is associated with categories **Mosaics** and **Justinian I**. \square

3.3 Path

Path is the description of the logical place of a node in a structure, and of how a user navigated to that node. In a hierarchy, there is only one path to each

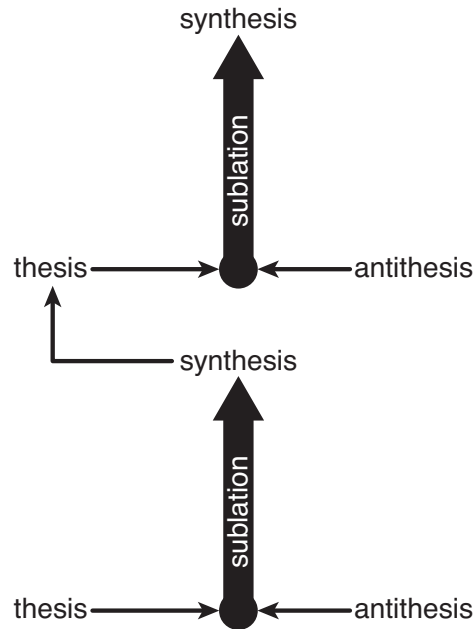


Fig. 9: a formulation of the dialectical process.

node. For instance, consider categories X and Y , and path X/Y . There is only one path X/Y to reach that node. In an organic hierarchy, however, many paths are possible because the order is inconsequential. In our example, X/Y and Y/X are equivalent because each path represents the intersection $X \cap Y = Y \cap X$. This flexibility of path is especially important for information discovery. It allows a user to navigate “up” along a different path than they navigated “down,” for instance.

4 Dialectic

The Fichtean/Hegelian *dialectic* provides a framework for the evolution of thinking. One formulation of this process is shown in Fig. 9. It begins with a *thesis*, which gives a perspective on a topic. If a new perspective is introduced, it is called an *antithesis*. Antitheses can be contradictory to the thesis, but often they are not. Rather, they reveal something previously unknown about the topic, and the relationship between the thesis and antithesis is unresolved. The process of *sublation* forms a *synthesis*, which includes both the thesis and antithesis. This synthesis is considered a new thesis. *Et cetera*.

4.1 Synthesis III: Structure, Flow, and Dialectic

In this final synthesis with that of Sect. 3.2, the dialectical framework can be used to express the evolution of thought. Each category may be assigned a

thesis flow that is a definition or description of the category.⁷ It will certainly intersect a number of other categories, but it is specifically used to define its *thesis category*. Similarly, any flow can be considered the thesis flow for a thesis category it generates.

When a flow is newly associated with a category, it becomes an *antithesis flow* of the category's thesis flow. It may contradict, supplement, or be redundant with the thesis flow. In any case, when a user sublates these flows, they become a *synthesis flow*, which is the new thesis flow for the category.

This third synthesis is called the *dialectical architecture*. It allows information to be structured as in the organic hierarchy (first synthesis), incorporates free-ranging flows (second synthesis), and systematically structures the evolution of thinking (third synthesis). This captures a third plane of information architecture.

Definition 12 (dialectic plane). The dialectic plane of information architecture is the evolution of the structures and flows within the architecture. \square

This plane is often ignored in information architecture and its instantiations. Attempts to incorporate it include *backup software*, *wiki* revision histories, and *version control*. Framing it with the dialectical method allows the user to focus on the evolution of the *ideas* and not a specific flow. It also allows the user to discover new antitheses that arise in connection with other work in the information system.

5 Conclusion and Prospects

We have presented a new information architecture, the dialectical architecture. It enables thinking to be structured in a general manner, flow through that structure, and evolve through dialectic — all in a computing medium.

As with other information architectures, there are many viable instantiations. One instantiation, a note-taking system, is currently in development. Much work remains to determine how successful the architecture is at providing a medium for human thinking. Its basis in the standard architectures, the method of loci, and the dialectical method is a reason for optimism.

References

1. Garrett, J.: Elements of User Experience, The: User-Centered Design for the Web and Beyond. Voices That Matter, Pearson Education (2010)
2. Maguire, E.A., Valentine, E.R., Wilding, J.M., Kapur, N.: Routes to remembering: the brains behind superior memory. *Nature Neuroscience* 6(1), 90 (2003)
3. Markoff, J.: What the Dormouse Said: How the Sixties Counterculture Shaped the Personal Computer Industry. Penguin Group (2005)

⁷ Each node may also be assigned a thesis flow. For instance, a thesis flow for category $X \cap Y$ might be *the relation of X and Y*.