Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2010), *24*, 259–276.
© Cambridge University Press, 2009 0890-0604/09 $25.00
doi:10.1017/S0890060409990102

Approximating shapes with hierarchies and topologies

DJORDJE KRSTIC
Calabasas, California, USA

(Received October 15, 2008; Accepted June 3, 2009)

Abstract

This is the second paper in the series on shape decompositions and their use as shape approximations. This time we investigate hierarchical and topological shape decompositions or hierarchies and topologies. We showed earlier that bounded decompositions behave the same way as shapes do. The same holds for hierarchies and topologies, which are special kinds of bounded decompositions. They are distinguished by their algebraic structures, which have many important properties to facilitate their application as shape approximations. We provide an account of their properties with an emphasis on their application.

Keywords: Hierarchies; Shape Algebras; Shape Decompositions; Shape Grammars; Topologies

1. INTRODUCTION

This is the second paper in the series on shape decompositions and their use as shape approximations. Some notions from the first paper (Krstic, 2004) and its extended version (Krstic, 2005) will be reiterated here for a good start.

Although shapes come unanalyzed rendering any division into parts possible, this is not how we usually perceive them. An attempt to understand or explain a shape leads inevitably to its decomposition into certain parts. Only the parts that are consistent with our understanding of the shape are recognized while all the other parts are neglected. Nevertheless, the recognized parts do account for the whole shape, which is their sum. Shape decompositions are therefore sets of shapes that emphasize certain properties of their sums. Decompositions are often used in place of the shapes serving as their approximations. How good are such approximations? This is an important question given the role shapes play not only in art and design but also in our everyday lives. We have addressed the question (Krstic, 2004, 2005) from the point of view of formal design theory featuring shape grammars and related shape algebras, where Stiny (2006) is the latest monograph on the subject.

This is an advantageous approach because shape grammars and shape algebras are both intuitive and rigorous tools. The grammars mimic the creative process of design where designers adopt and follow rules to create designs or break the rules only to create and follow new ones. The algebras support the rules by providing operations that formalize the designer's actions. The operations model what designers traditionally do when they draw or erase shapes, build, or modify models or move shapes around producing different spatial relations. In contrast to their intuitive side, shape grammars and algebras are rigorous tools grounded in mathematics. The latter informs both practical computations with shapes as well as the theoretical inquiry into their nature.

We rely on the mathematics of shapes to address the problem of shapes being represented by their decompositions. The problem stems from the fact that sets of entities and entities themselves are different in nature, which becomes even more evident in computations. Shapes and related shape algebras behave (in general) differently than shape decompositions and their algebras. This renders decompositions poor substitutes for shapes. In contrast, shape decompositions of certain structure do behave like shapes. In particular, bounded decompositions behave in computations the same way shapes do. Their algebras and shape algebras are isomorphic.

Bounded decompositions recognize the shape they analyze and the empty shape and may contain other shape parts besides the two. At minimum, no other parts are included. The structure of a bounded decomposition puts its elements into certain contexts.

A local context, which assures that the shape parts are always seen in relation to the whole, is established by the fact that a bounded decomposition always includes the shape it analyzes.

A global context, in which the attention is shifted from shape parts to shape surroundings, is given by the presence of the empty shape. The absence of shape parts signified by the empty shape implies the shape surroundings.

Local and global contexts are both necessary and sufficient for decompositions to be successful shape approximations.

Reprint requests to: Djordje Krstic, 22415 Kearny Street, Calabasas, CA 91302, USA. E-mail: gkrstich@aol.com

As there are no additional requirements, any shape part can easily be included into a bounded decomposition if needed. Consequently, bounded decompositions may serve as the basis for other structured decompositions. They may satisfy additional conditions to become lattices, hierarchies, topologies, and Boolean algebras. All of these structures may be used as shape approximations for different purposes.

We will explore hierarchical and topological decompositions and their use in design.

1.1. Background

The symbols $\emptyset$, $\cup$, $\cap$, –, $\in$, $\notin$, $\subseteq$, and $\wp$ will be used in their standard set-theoretic meaning as the empty set, union, intersection, difference, element of, not an element of, subset of, and power set, respectively. Some standard set-theoretic notions will be assumed, such as the ordered set with related maximal and minimal elements, bounds, and order-preserving mappings as well as the direct product and related functions and relations.

We will see topologies and hierarchies for shapes as *lattices*. These are partially ordered sets turned algebras with two binary operations: *meet* and *join*. The operations are, respectively, denoted by $\vee$ and $\wedge$, and defined as the greatest lower bound and the least upper bound of the arguments. Both bounds have to be defined for every pair of elements of a set in order for it to be a lattice. If only one of the bounds is defined then the set is a *semilattice* and may be a *join-semilattice* or a *meet-semilattice*.

By the *duality* principle any expression valid for a lattice is also valid if $\vee$ and $\wedge$ exchange places. A lattice may have two distinguished elements 0, which is the smallest element, and its dual 1, which is the greatest element. If a lattice has 0 then the meet of any of its elements with 0 is 0. Dually in lattices with 1 the join of any element with 1 is 1.

When defining hierarchies for shapes we rely on the concept of reducibility in lattices. An element of a lattice is *reducible* if when removed it may be reconstructed as a meet or a join of some of the remaining elements. In particular, an element $\boldsymbol{x}$ is *meet-reducible* if $\boldsymbol{x} = \boldsymbol{y} \wedge \boldsymbol{z}$ and $\boldsymbol{y}, \boldsymbol{z} > \boldsymbol{x}$, where $\boldsymbol{y}$ and $\boldsymbol{z}$ are elements. Dually, $\boldsymbol{x}$ is *join-reducible* if $x = \boldsymbol{y} \vee \boldsymbol{z}$ and $\boldsymbol{y}, \boldsymbol{z} < \boldsymbol{x}$.

Special kinds of order-preserving mappings play an important role in both hierarchies and topologies for shapes. These are *closures* and *interiors*, which are *operators* or mappings of a set to itself. Closure Γ: $A \rightarrow A$ defined on a partially ordered set A, with x as an element, is *idempotent* and *extensive* or $\Gamma(\Gamma(x)) = \Gamma(x)$ and $x \leq \Gamma(x)$, respectively. In contrast, interior Ω: $A \rightarrow A$ is *idempotent* and *intensive* or $\Omega(x) \leq x$.

If A is a lattice with 0, Γ may satisfy additional conditions to become a *topological* closure. That is, $\Gamma(x) \vee \Gamma(y) = \Gamma(x \vee y)$ and $\Gamma(0) = 0$, where x, $y \in A$. Dually, if A has 1, Ω is a topological interior if $\Omega(x) \wedge \Omega(y) = \Omega(x \wedge y)$ and $\Omega(1) = 1$.

The image of Γ is a set of closed elements of A, whereas the image of Ω is the set of open elements of A.

We also make use of *shape algebras* (Stiny, 1991, 1992, 2006; Krstic, 1999, 2001). These feature Boolean operations for shapes and are also closed under geometric transformations. Shape algebra $\boldsymbol{U_{ij}}$ computes with i-dimensional shapes occupying j-dimensional space, where $i, j = 0, 1, 2, 3$ and $i \leq j$. It has a lower bound, which is the empty shape denoted by 0, but has no upper bound. Shapes from $\boldsymbol{U_{ij}}$, together with Boolean operations of sum $+$, product $\cdot$, and difference $-$, form a *generalized Boolean algebra*, a relatively complemented distributive lattice with a smallest element (Birkhoff, 1993). A compound operation of *symmetric difference* $\oplus$ may be defined on $\boldsymbol{U_{ij}}$, so that $\boldsymbol{x} \oplus \boldsymbol{y} = (\boldsymbol{x} - \boldsymbol{y}) + (\boldsymbol{y} - \boldsymbol{x}) = (\boldsymbol{x} + \boldsymbol{y}) - (\boldsymbol{x} \cdot \boldsymbol{y})$, where $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{U_{ij}}$. It corresponds to exclusive or (*XOR*) operation in Boolean logic. Symmetric difference has somewhat conflicting nature: it is a sum for discrete shapes and a difference for comparable ones.

The set of all parts of a shape $\boldsymbol{a}$ or its *maximal structure* denoted by $\mathbf{B}(\boldsymbol{a})$ is a proper subalgebra of $\boldsymbol{U_{ij}}$ closed under the symmetry group of $\boldsymbol{a}$ (Earl, 1997; Krstic, 2004, 2005; Stiny, 2006).

Finally, *shape decomposition algebras* provide the framework for hierarchies and topologies for shapes. There are different families of such algebras (Krstic, 2004, 2005), but we only use complex algebras of decompositions. These are extensions of shape algebras $\boldsymbol{U_{ij}}$ to finite sets of shapes and are denoted by $\wp(\boldsymbol{U_{ij}})$. Their operations are extensions of shape operations to direct products of decompositions and are defined as $A + B = +(A \times B)$, $A \cdot B = \cdot(A \times B)$, and $A - B = -(A \times B)$, where A and B are decompositions and $*(A \times B) = \{x * y |\ x \in A, y \in B\}$, where $*$ stands for a binary operation.

1.2. Structured decompositions

The structure of a decomposition may be seen on two levels: local and global. On the local level the decomposition is a set of shapes that puts emphasis on the relations among its elements. In contrast, on the global level the decomposition is seen as analyzing a shape, the sum of its elements, so that the relations between parts of the shape and elements of the decomposition are exposed. If a decomposition is to be an approximation of a shape, then its structure on a global level is of the most importance.

We should, at minimum, require for such a decomposition to have a representation for each part of the shape it analyzes. There are infinitely many parts, but only finitely many elements of the decomposition so that the relation between the two is many to one.

For shape $\boldsymbol{a}$, its part $\boldsymbol{x}$, and its decomposition A this relation is as follows. If there are elements u and $\boldsymbol{v}$ in A such that $\boldsymbol{u} \leq \boldsymbol{x} \leq \boldsymbol{v}$, then A approximates $\boldsymbol{x}$ so that $\boldsymbol{x}$ is *bottomed* by $\boldsymbol{u}$, *topped* by $\boldsymbol{v}$, and *bounded* by both $\boldsymbol{u}$ and $\boldsymbol{v}$. Part $\boldsymbol{x}$ has at least properties of $\boldsymbol{u}$ and at most properties of $\boldsymbol{v}$. If neither $\boldsymbol{u}$ nor $\boldsymbol{v}$ exists, then $\boldsymbol{x}$ is not recognized by A. Elements of A are bounded by themselves, or if $\boldsymbol{x} \in A$ then $\boldsymbol{u} = \boldsymbol{x} = \boldsymbol{v}$. If $\boldsymbol{x}$ is bounded by more than one pair of elements of A and it is impossible to

https://doi.org/10.1017/S0890060409990102 Published online by Cambridge University Press

pick one of the pairs as the representative of $\boldsymbol{x}$, then its representation is ambiguous.

For example, bounded decompositions satisfy the minimum requirement above because each element of $\boldsymbol{a}$ is represented (trivially) in A by the pair of shapes 0 and $\boldsymbol{a}$. However, if A has more than two elements then the representation of some parts of a may be ambiguous.

There is no ambiguity in the relation of a shape to its parts, and there should be no ambiguity in how the parts are represented by a decomposition that approximates the shape. This calls for the sharpening of the requirement above by making it the *unique representation requirement*, a decomposition that serves as a shape approximation should have the unique representation for each part of the shape it analyzes.

Decompositions of certain structures satisfy the unique representation requirement above, chief among which are the decompositions that have structures of hierarchies and topologies.

2. HIERARCHIES

Hierarchies are well-known structured decompositions that are widely used in arts and sciences for categorizing and organizing large numbers of objects. Taxonomies, such as biological taxonomies, are hierarchies and so are many databases and file systems.

Hierarchies are ordered sets with unique paths between their elements.[1] In design, they distinguish parts and show how these are put together to assemble shapes. A shape may be decomposed in a hierarchical fashion with the aid of a simple recursive procedure, which breaks down shapes into their discrete parts. It starts by breaking a shape and proceeds by breaking its parts and then the parts of the parts and so on. The result is a treelike structure with the original shape at the top and the minimum elements or *atoms* at the bottom. Hierarchies describe designs and their components as sums of atoms.

For example, the hierarchy in Figure 1 shows how a shape, representing an IKEA coffee table, is progressively broken down into discrete parts representing the table legs and the board, which are the atoms of the hierarchy. It also shows how this shape can be assembled from the atoms. Hierarchies are easy to use, but they may not be the best solution for every application.

2.1. Definition

The underlying schema (Hasse diagram) in Figure 1 is typical of hierarchical decompositions of shapes and its inspection may reveal their structure.

We notice that the hierarchy contains the least upper bound for every pair of its elements, but lacks the greatest lower bound. This renders hierarchical decompositions join-semilattices with a greatest element, which is the shape analyzed by the hierarchy.

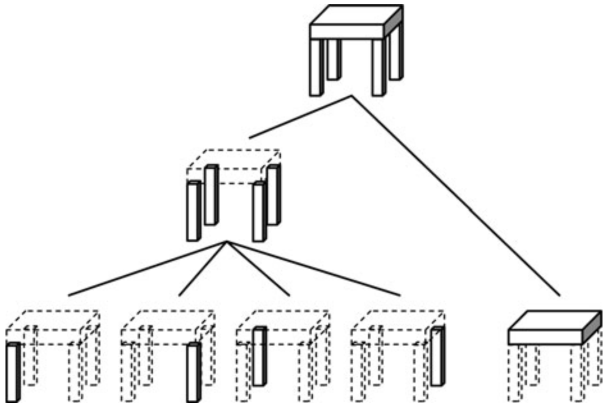

Fig. 1. The hierarchy of parts of an IKEA coffee table.

Each element of the hierarchy, with the exception of its atoms, covers at least two elements, which renders such elements join-reducible. In contrast, each element, with the exception of the greatest element, is covered by exactly one element, which means that only comparable elements have meets.

For every nonempty part of the coffee table in Figure 1 there is an element in the hierarchy that uniquely represents it. This element is the least upper bound of the part. For example, in Figure 2 shape (a) is a part of the coffee table while shape (b) is its least upper bound with respect to the hierarchy in Figure 1.

Although nonempty parts are uniquely represented, the hierarchy does not satisfy the unique representation requirement. It lacks a unique element to top the empty shape, which is part of every shape. Every atom of the hierarchy is as good of a representation of the empty shape as any other atom. This may be rectified by including the empty shape in the hierarchy. Now the empty shape represents itself and the hierarchy satisfies the unique representation requirement.

The inclusion of the empty shape has some interesting consequences on the structure of the hierarchy. Now every pair of noncomparable elements in the hierarchy has a meet that is the empty shape. Pairs of comparable elements already have meets, as noted earlier, so that the hierarchy is closed under the meet operation. The structure of the hierarchy changes from a semilattice to a lattice. This rather unusual representation (hierarchies are usually considered trees or join-semilattices) offers several advantages:

1. Because hierarchies contain the shape they analyze and (now) the empty shape they are bounded decompo-

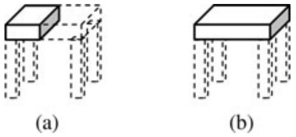

Fig. 2. (a) A part of an IKEA coffee table and (b) its representation in the hierarchy.

[1] Hierarchies are viewed as ordered sets rather than tree structures, in a set theoretic rather then graph theoretic fashion. The latter view may be more general, but the former has an advantage of exposing (partial) order, which is central to shape decompositions.

https://doi.org/10.1017/S0890060409990102 Published online by Cambridge University Press

sitions. The latter are known to behave like shapes and form an algebra isomorphic with an appropriate shape algebra (Krstic, 2004, 2005).

2. Hierarchies (now) have the unique representative for the empty shape and may satisfy the unique representation requirement to qualify as shape approximations.
3. Hierarchies are (now) lattices and so are topologies, which we will show later, so that a unique characterization of both is possible.

We now give a formal definition of a hierarchy.

Definition 1. A bounded decomposition of shape $\boldsymbol{a}$ is a *hierarchical decomposition* of $\boldsymbol{a}$ or a *hierarchy* on $\boldsymbol{a}$ denoted by $X(\boldsymbol{a})$ if it is a lattice with all nonempty elements being meet-irreducible and nonempty elements other than atoms being join-reducible. ■

The meet-irreducibility condition assures that no nonempty element of a hierarchy is a part of two disjoint components of the design. As a consequence, between any two (comparable) components of the design there is a unique path, traversing nonempty elements of the hierarchy, which means that the design is assembled in a unique way. In contrast, the join-reducibility condition is necessary to make sure that proper parts of any component of the design when assembled constitute precisely that component. This condition also prevents chains from being mistaken for hierarchies.

Note that inclusion of 0 is formal in nature and we omit 0 in diagrams in order to maintain the familiar look of hierarchies. We also omit 0 in element counts.

2.2. Properties of hierarchical decompositions of shapes

Hierarchies are rich in properties, but we will only examine those that are relevant to design. Prime among such properties is the ability to satisfy the unique representation requirement, which qualifies hierarchies as shape approximations.

We observed that the hierarchy in Figure 1 satisfies the requirement, but not all hierarchies have this property. Many hierarchies that serve to order descriptions or properties of designs do not have unique representations for design parts.

For example, a hierarchy in Figure 3 describes a small house consisting of a kitchen, living room, hallway, bedroom, bathroom, walk-in closet, stairway, and garage. In this hierarchy of names, which distinguish spaces, elements combine disjunctively without a conjunctive possibility. The bedroom has no common part with the bathroom or walk-in closet, although together they are the private spaces of the house. The same is true for the living room, hallway, kitchen, stairway, and garage, which are the public spaces. If we replace the names with the shapes they represent, the conjunction of elements become possible, but the hierarchy is oblivious to that.

For example, the hierarchy in Figure 3 may represent the Ohlenbusch residence in San Antonio, Texas, designed by Darryl Ohlenbusch Design and completed in 1995 (Ojeda, 1997, p. 128). Its plan in Figure 4a appears as a top element of the hierarchy, and spaces d–k representing the stairway, garage, kitchen, bathroom, living room, bedroom, closet, and hallway, respectively, are the atoms of the hierarchy. Disjunctive combinations of atoms form the public space of the house as well as its private space (Fig. 4b and 4c, respectively). This enumerates all of the elements of the hierarchy as prescribed by the schema (Hasse diagram) in Figure 3. What is not prescribed by the schema is the possibility for conjunctive combinations. The kitchen and living room may combine (conjunctively) to reveal the walls they share. The same is true for other adjacent spaces like the kitchen and bathroom or the bathroom and bedroom. For example, the shape in Figure 5c is the wall between the kitchen (b) and bathroom (a). It is topped by these two shapes and also by some other elements of the hierarchy that include the two as parts. Because the kitchen and bathroom are noncomparable shapes, the shape in Figure 5c has two representatives, which violates the unique representation requirement.

Only hierarchies of certain structure satisfy the requirement in accordance with the following result.

Proposition 1. *A hierarchy satisfies the unique representation requirement if and only if its meet operation is the product from the underlying shape algebra.*[2] ■

Let $\boldsymbol{a}$ be a shape defined in $\boldsymbol{U}_{ij}$, $\boldsymbol{x}$ its part, and $X(\boldsymbol{a})$ a hierarchical decomposition of $\boldsymbol{a}$. If $X(\boldsymbol{a})$ satisfies the unique representation requirement then there is a shape $\boldsymbol{y} \in X(\boldsymbol{a})$ which represents $\boldsymbol{x}$ so that either $\boldsymbol{x} \leq \boldsymbol{y}$ or $\boldsymbol{y} \leq \boldsymbol{x}$. Because $X(\boldsymbol{a}) - \{0\}$ is a join-semilattice, the former inequality holds and because $\boldsymbol{y}$ represents $\boldsymbol{x}$ uniquely, $\boldsymbol{y}$ is the least upper bound of $\boldsymbol{x}$ in $X(\boldsymbol{a})$. Consequently, $X(\boldsymbol{a})$ has the least upper bound for every part of $\boldsymbol{a}$, which requires for $X(\boldsymbol{a})$ to be closed under the product from $\boldsymbol{U}_{ij}$. The product is, therefore, the meet operation of $X(\boldsymbol{a})$. In contrast, if the meet from $X(\boldsymbol{a})$ is the product from $\boldsymbol{U}_{ij}$ then $X(\boldsymbol{a})$ is closed under the product, which defines a closure Γ on the set $\mathbf{B}(\boldsymbol{a})$ of all parts of $\boldsymbol{a}$ such that $\Gamma(\mathbf{B}(\boldsymbol{a})) = X(\boldsymbol{a})$. Part $\boldsymbol{x}$ is uniquely represented by its closure $\Gamma(\boldsymbol{x})$, which is an element of $X(\boldsymbol{a})$ so that $X(\boldsymbol{a})$ satisfies the unique representation requirement.

Note that Γ is not a topological closure. Take, for example, a four-element hierarchy satisfying Proposition 1 and consisting of three atoms $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ and their sum $\boldsymbol{a}$. The sum of closures $\Gamma(\boldsymbol{x}) + \Gamma(\boldsymbol{y}) = \boldsymbol{x} + \boldsymbol{y}$, is different from the closure of the sum $\Gamma(\boldsymbol{x} + \boldsymbol{y}) = \boldsymbol{a} = \boldsymbol{x} + \boldsymbol{y} + \boldsymbol{z}$ so that Γ is not a topological closure.

Hierarchies that satisfy Proposition 1 have atoms that are pairwise discrete. This is a corollary of Proposition 1: if $\boldsymbol{x}$ and $\boldsymbol{y}$ are atoms then $\boldsymbol{x} \wedge \boldsymbol{y} = \boldsymbol{x} \cdot \boldsymbol{y} = 0$, hence, they are discrete.

[2] That is, the algebra in which the elements of the hierarchy are defined.

https://doi.org/10.1017/S0890060409990102 Published online by Cambridge University Press

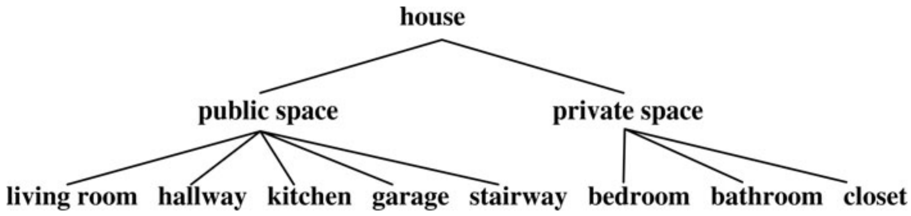

Fig. 3. The hierarchy of spaces for a small house.

The set of atoms of such a hierarchy is a discrete decomposition of the shape analyzed by the hierarchy. This motivates the following definition.

Definition 2. Hierarchical decompositions with meets coinciding with products are *discrete hierarchies*. ■

We may now reformulate Proposition 1.

Corollary 1. *A hierarchy satisfies the unique representation requirement if and only if it is discrete.* ■

Meet operation of a discrete hierarchy is uniquely determined by the shapes that are its arguments. It equals their product, which remains the same regardless of the hierarchy the shapes belong to. This is not the case with the join operation. The latter is the least upper bound determined by its arguments, but also by all other elements of the hierarchy. For a discrete hierarchy $X(\boldsymbol{a})$ with elements $\boldsymbol{x}$ and $\boldsymbol{y}$ meet is given by $\boldsymbol{x} \wedge \boldsymbol{y} = \boldsymbol{x} \cdot \boldsymbol{y}$ and join by $\boldsymbol{x} \vee \boldsymbol{y} = \Gamma(\boldsymbol{x} + \boldsymbol{y})$, where Γ is a closure operator on $\mathbf{B}(\boldsymbol{a})$ such that $\Gamma(\mathbf{B}(\boldsymbol{a})) = X(\boldsymbol{a})$. A discrete hierarchy is defined by its join. We can have a kit of parts from which we assemble an object and we can do it in many different ways. Each of the ways defines a discrete hierarchy with a different join operation. The sizes of such hierarchies range from $k + 1$ to $2k - 1$, where k is the number of parts in the kit. The former is the size of a hierarchy constructed by just adding the sum of atoms as a top element. This hierarchy may then be altered by adding a shape that covers (at most) two of the existing shapes. If this addition is repeated exhaustively, the resulting hierarchy will have the maximum size.

If a hierarchy is used to describe the materialization of a design then its elements are components of the design. Each component should be accounted for by smaller components that are its proper parts. Hierarchies, by definition, support this because every nonempty element of a hierarchy, with exception of the atoms, is the join of other elements that are its proper nonempty parts. Although necessary, this is not a sufficient condition as there are hierarchies with elements that are not accounted for by their proper parts.

For example, hierarchy in Figure 6 shows an IKEA coffee table with its two typical parts: a board and leg. The join of the two parts is the table itself, which leaves the three legs of the table unaccounted for. In order to avoid this, the components should be sums of smaller components, which leads to a special type of hierarchy in accordance with to the following definition.

Definition 3. Shape $\boldsymbol{x}$ is a *sum-reducible* element of a hierarchical decomposition $X(\boldsymbol{a})$ if there is a subset $S \subseteq X(\boldsymbol{a})$ such that $\boldsymbol{x}$ is not an element of S and $\boldsymbol{x}$ is the sum of the elements of S. A hierarchy is *summable* if all of its join-reducible elements are sum-reducible. ■

It is clear that elements of S are proper parts of $\boldsymbol{x}$, because $\boldsymbol{x} = \Sigma S$, $\boldsymbol{x} \notin S$, and a sum is greater or equal to any of its arguments, so that sum-reducible elements are also join-reducible. The converse does not hold, as in hierarchy in Figure 6. The difference between join-reducibility and sum-reducibility deserves a closer look.

For example, let shape $\boldsymbol{a}$ be analyzed by a discrete four-element hierarchy with three atoms $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ such that $\boldsymbol{a}$ is the sum of the atoms. Shape $\boldsymbol{a}$ is both join-reducible and sum-reducible, however, where subsets $\{\boldsymbol{x}, \boldsymbol{y}\}$, $\{\boldsymbol{x}, \boldsymbol{z}\}$, $\{\boldsymbol{y}, \boldsymbol{z}\}$, and $\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\}$ have $\boldsymbol{a}$ as the join only the last subset has it as the sum. If we change the atoms so that $\boldsymbol{x} \cdot \boldsymbol{y} \neq 0$ and $\boldsymbol{z} = \boldsymbol{x} \oplus \boldsymbol{y}$, subset $\{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}\}$ still has $\boldsymbol{a}$ as the sum, but so do the remaining three subsets.

Join-reducibility is a general property because it depends only on the partial order among the elements involved. It is enough to inspect the Hasse diagram of a hierarchy to decide if an element is join-reducible. In contrast, sum-reducibility is an individual property because it also depends on the elements themselves.

The hierarchy in Figure 1 is summable and this should be true of any hierarchy describing the materialization of a design. We start with the smallest components of the design and build larger components from them and even larger components from these components and continue the process until the design is complete. Consequently, every component of the design consists of the smallest components that we started with, which leads to the following proposition.

Proposition 2. *A nonempty element of a hierarchy is a join of atoms. It is also a sum of atoms if the hierarchy is summable.* ■

The proposition (trivially) works for atoms because $\boldsymbol{a} \vee \boldsymbol{a} = \boldsymbol{a}$ and $\boldsymbol{a} + \boldsymbol{a} = \boldsymbol{a}$, where $\boldsymbol{a}$ is an atom. In accordance with Definition 1 a nonempty element of a hierarchy is either an atom or it covers, and it is a join of at least two and possibly more elements of the hierarchy. Some of these elements may be atoms, but the other ones are themselves joins of the elements they cover. Because join is an associative operation we may substitute these elements with the related joins of elements. We may continue with such substitutions until there

https://doi.org/10.1017/S0890060409990102 Published online by Cambridge University Press

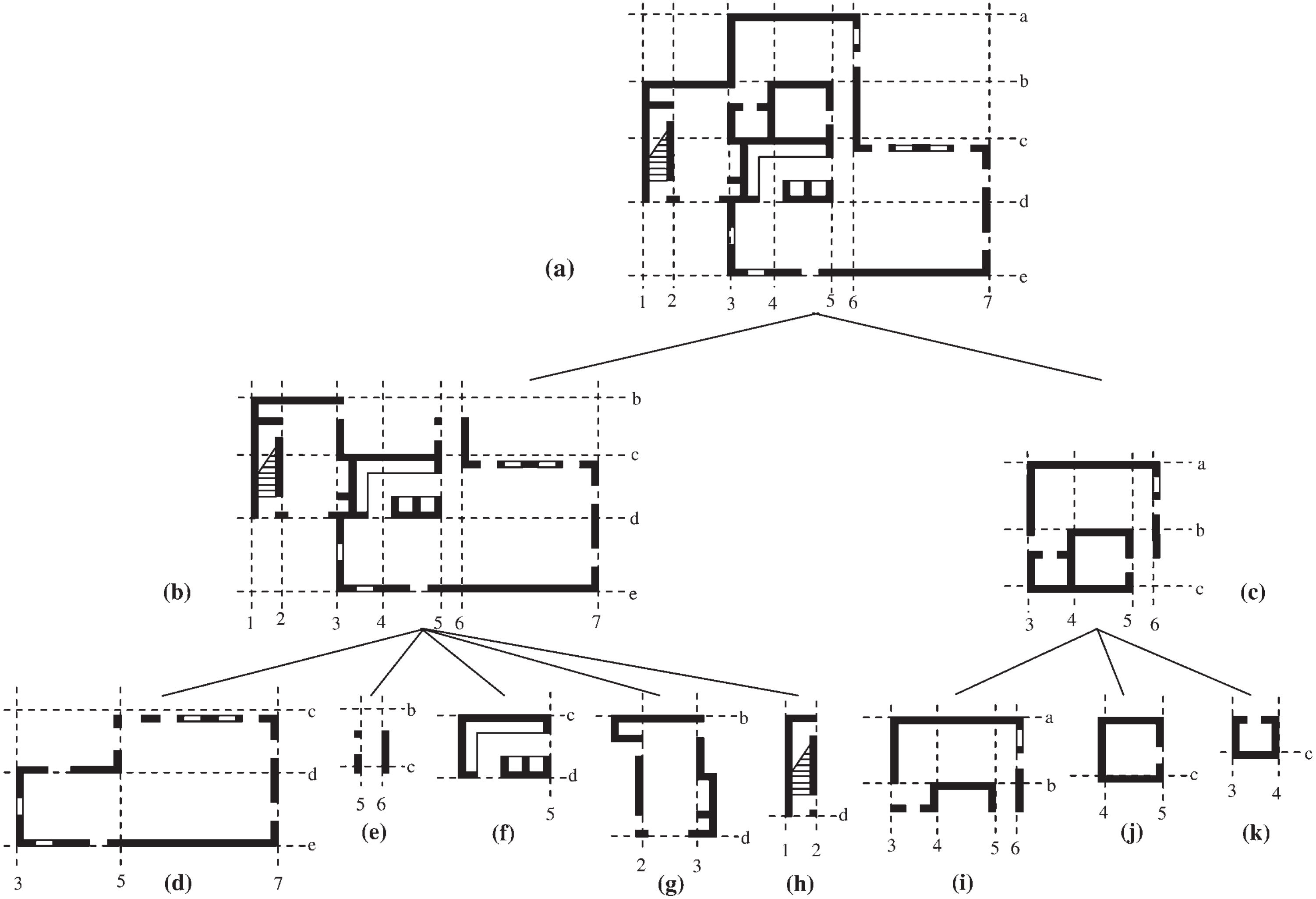

Fig. 4. The Ohlenbusch residence: (a) plan, (b) public space, (c) private space, (d) stairway, (e) garage, (f) kitchen, (g) bathroom, (h) living room, (i) bedroom, (j) closet, and (k) hallway.

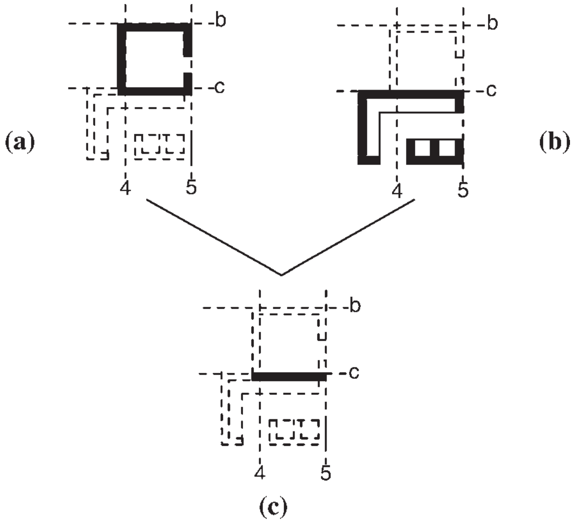

Fig. 5. The Ohlenbusch residence: (a) bathroom, (b) kitchen, and (c) a partitioning wall between the two.

are no more elements that cover other nonempty elements. At this point all of the elements are atoms and our original element is a join of atoms of the hierarchy. According to Definition 3, in summable hierarchies all of the join-reducible elements are sum-reducible, and we may use the same procedure with sums instead of joins.

The fact that the top element of a hierarchy is the join of all of its atoms comes as a corollary of the proposition above. By the same token, the top element is the sum of the atoms if the hierarchy is summable.

We conclude this account with an interesting property that unites the concepts of summability and discreteness.

In summable hierarchies an element is the sum of the elements it covers and if the hierarchy is also discrete these elements are pairwise discrete. This creates an opportunity to replace the sum of elements with their symmetric difference, because symmetric difference of discrete shapes is equal to their sum or if $\boldsymbol{x} \cdot \boldsymbol{y} = 0$ then $\boldsymbol{x} \oplus \boldsymbol{y} = \boldsymbol{x} + \boldsymbol{y}$. Definition 3 may now be reformulated as follows.

Definition 4. Shape $\boldsymbol{x}$ is a *symmetric-difference-reducible* or *XOR-reducible* element of a hierarchy if $\boldsymbol{x}$ is the symmetric difference of the elements it covers. A hierarchy is *XOR* if all of its join-reducible elements are XOR-reducible. ■

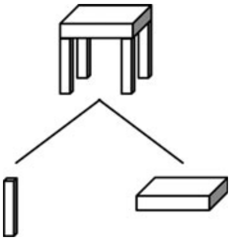

Fig. 6. The typical parts of an IKEA coffee table.

To see how Definition 4 works let us assume that $\boldsymbol{x}$ covers two elements $\boldsymbol{y}$ and $\boldsymbol{z}$ and that $\boldsymbol{x} = \boldsymbol{y} \oplus \boldsymbol{z}$. If $\boldsymbol{y} \cdot \boldsymbol{z} = 0$ then $\boldsymbol{x} = \boldsymbol{y} \oplus \boldsymbol{z} = \boldsymbol{y} + \boldsymbol{z}$ and $\boldsymbol{x}$ is XOR-reducible. In contrast, if $\boldsymbol{y} \cdot \boldsymbol{z} \neq 0$ then $\boldsymbol{x} = \boldsymbol{y} \oplus \boldsymbol{z} = (\boldsymbol{y} + \boldsymbol{z}) - (\boldsymbol{y} \cdot \boldsymbol{z})$ so that neither $\boldsymbol{y}$ nor $\boldsymbol{z}$ are parts of $\boldsymbol{x}$, which contradicts the original assumption that $\boldsymbol{x}$ covers $\boldsymbol{y}$ and $\boldsymbol{z}$. This may be summarized in the following proposition.

Proposition 3. *A hierarchy is summable and discrete if and only if it is XOR.* ■

Replacing sum with symmetric difference has a practical side: it allows for simultaneous calculations with both shapes and their boundaries, which is discussed in the next section.

2.3. Applications and constructions of hierarchies

Hierarchies that are summable and discrete are well suited for applications in design. We should explore possibilities of their use and construction with an eye on their role as shape approximations.

An XOR hierarchy is an excellent choice for representing the materialization of a design. It has a useful property of considering parts disjoint only if they are physically discrete. Such realism on the part of representation allows for an object to be really built out of parts that are the atoms of the hierarchy analyzing the object. In particular, both the object and its components are sums of atoms of the hierarchy. The atoms are pairwise discrete which prevents collisions of components and makes the hierarchy an excellent representation for the object assembling process. Finally, being discrete guarantees that the hierarchy satisfies the unique representation requirement that makes it a good shape approximation.

An XOR hierarchy has the least upper bound for every part of the object it analyzes, which renders it an excellent representation for the purpose of maintaining the object. For example, consider the hierarchy in Figure 1 as a set of replaceable parts of the table, which, just in case of a total loss, includes the table itself. Now suppose that the part in Figure 2a gets damaged by termites. The smallest among the replaceable parts that contain part (a) should be replaced in order to offset the damage. That is the part in Figure 2b that is the closure of part (a) or its least upper bound with respect to the hierarchy. This representation may be improved by assigning a set of attributes to each element of the hierarchy. Each attribute may describe a function the element performs. The sets of attributes form a hierarchy isomorphic to the original one. If the object is not performing certain functions, the smallest replaceable part capable of performing these functions should be replaced. To find the part one takes the closure of the malfunctioning attributes in the hierarchy of attribute sets and then maps it, via isomorphism, to the hierarchy of replaceable parts.

If an XOR hierarchy is considered a shape approximation, then its size matters. In hierarchy $X(\boldsymbol{a})$ a part $\boldsymbol{x}$ of $\boldsymbol{a}$ is represented by its closure $\Gamma(\boldsymbol{x}) \in X(\boldsymbol{a})$, but this relation is many to one, there are (infinitely) many parts of $\boldsymbol{a}$ represented by

https://doi.org/10.1017/S0890060409990102 Published online by Cambridge University Press

the same element of $X(\boldsymbol{a})$. This calls for an equivalence relation $\equiv$ on parts of $\boldsymbol{a}$ such that $\boldsymbol{x} \equiv \boldsymbol{y}$ if $\Gamma(\boldsymbol{x}) = \Gamma(\boldsymbol{y})$. This relation partitions the parts of $\boldsymbol{a}$ into equivalence classes corresponding to nonempty elements of $X(\boldsymbol{a})$. The more elements a hierarchy has the more equivalence classes there are and the better it approximates the shape. However, not all of the parts of $\boldsymbol{a}$ are represented equally well in $X(\boldsymbol{a})$. Parts that are elements of $X(\boldsymbol{a})$ are best represented. In contrast, *dense* parts, or parts with closures that equal $\boldsymbol{a}$, may be poorly represented by the hierarchy. This is especially true of small dense parts where their size is in disagreement with the fact that only $\boldsymbol{a}$ is big enough to account for all of their properties.

For example, a dense part may be constructed by picking an arbitrarily small part from every atom of $X(\boldsymbol{a})$ and then summing these parts. The resulting part can be arbitrarily small, but its closure is $\boldsymbol{a}$; thus, when it breaks down the whole object analyzed by the hierarchy is a total loss. If the table from our previous example is damaged so that a tiny termite damage appears on each of the legs and the board, the total damage is small, but the table is a total loss.

Shapes are approximated by their decompositions but also by their boundaries. Being one dimension lower than the shapes they delineate, boundaries are simple and economical representations. For example, in solid modeling, surface-based models represent solids indirectly via their bounding surfaces, which are also used to calculate the engineering properties of solids (Timmer & Stern 1980; Lee & Requicha, 1982). Likewise, in mathematics, many properties of three-dimensional (3-D) shapes are expressed via volume integrals. The well-known method of direct integration may then be used to represent a volume integral over a shape via the sum of simpler surface integrals over its boundary. The closely related divergence theorem of vector calculus yields a similar result.

Given the importance of shape boundaries it would be advantageous for any decomposition acting as a shape approximation to take account of the boundaries of its elements. XOR hierarchies are well suited for that because, in accordance with Definition 4, their reducible elements are symmetric-difference reducible and symmetric difference has an interesting property that ties shapes and their boundaries.

PROPOSITION 4. *The boundary of the symmetric difference of diagonal shapes is the symmetric difference of their boundaries (Earl, 1997; Krstic, 2001).* ■

Proposition 4 does not apply to all of the shapes but to a special class of shapes: diagonal shapes. The latter are defined in an algebra $\boldsymbol{U}_{ii}$ where shapes and the space in which they are manipulated are of the same dimension $i = 0, 1, 2$, or 3. Such are the solids in 3-D space or planar segments in a plane, which are expressive enough for most design (engineering) applications. The former pertain to the models designers build, including 3-D computer-aided design models, while the latter relate to 2-D drawings, which are traditional representations in design.

Figure 7 illustrates Proposition 4. XOR hierarchy (a) analyzes a double square defined in U_{22} algebra, which manipulates planar segments in a plane. The symmetric difference of the two squares, which are the atoms of the hierarchy, yields the double square and so does their sum. In contrast, the symmetric difference of the boundaries of the atoms defined in $\boldsymbol{U}_{12}$, which manipulates lines in the plane, yields the boundary of the double square (b), but their sum, shape (c), is greater than the boundary, because it contains a line that divides the double square.

To account for both shapes and their boundaries a hierarchy has to be XOR with elements that are ordered pairs of shapes paired with their boundaries. The pairs are combined with a single operation of symmetric difference in the framework of $\boldsymbol{UB}_i$ algebra ($i = 1, 2$, or 3), which is closed under Euclidean transformations that act on the pairs (Krstic, 2001). Note that because points have no boundaries there is no $\boldsymbol{UB}_0$ algebra. The operation is applied componentwise in the following way.

Let b be an operator that takes shape $\boldsymbol{a}$ defined in $\boldsymbol{U}_{ij}$ to its boundary $b(\boldsymbol{a})$ in $\boldsymbol{U}_{ij-1}$ and let $\boldsymbol{x}$ and $\boldsymbol{y}$ be diagonal shapes so that pairs $(\boldsymbol{x}, b(\boldsymbol{x}))$ and $(\boldsymbol{y}, b(\boldsymbol{y}))$ are defined in a $\boldsymbol{UB}_i$ algebra. The symmetric difference of the two pairs is

$$(\boldsymbol{x}, b(\boldsymbol{x})) \oplus (\boldsymbol{y}, b(\boldsymbol{y})) = (\boldsymbol{x} \oplus \boldsymbol{y}, b(\boldsymbol{x}) \oplus b(\boldsymbol{y})).$$

Because $\boldsymbol{x}$ and $\boldsymbol{y}$ are diagonal shapes Proposition 4 applies to yield

$$(\boldsymbol{x}, b(\boldsymbol{x})) \oplus (\boldsymbol{y}, b(\boldsymbol{y})) = (\boldsymbol{x} \oplus \boldsymbol{y}, b(\boldsymbol{x} \oplus \boldsymbol{y})).$$

If $\boldsymbol{x}$ and $\boldsymbol{y}$ belong to an XOR hierarchy, and are covered by the same element, then they are discrete and symmetric difference has the effect of a sum. The computation above is as

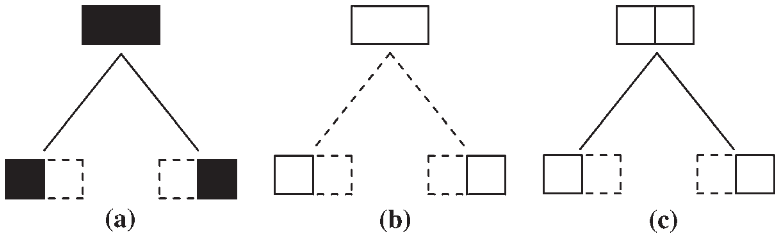

Fig. 7. The sum and symmetric difference hierarchy: (a) with the top element being the sum or the symmetric difference of the atoms, (b) the symmetric difference, and (c) the sum of the boundaries of the atoms.

https://doi.org/10.1017/S0890060409990102 Published online by Cambridge University Press

follows:

$$(\boldsymbol{x}, b(\boldsymbol{x})) + (\boldsymbol{y}, b(\boldsymbol{y})) = (\boldsymbol{x}, b(\boldsymbol{x})) \oplus (\boldsymbol{y}, b(\boldsymbol{y})) \\ = (\boldsymbol{x} \oplus \boldsymbol{y}, b(\boldsymbol{x} \oplus \boldsymbol{y})) = (\boldsymbol{x} + \boldsymbol{y}, b(\boldsymbol{x} + \boldsymbol{y})).$$

The computation starts with a sum, as if modeling an assembling process, which is then recast as the symmetric difference in order to preserve shape boundaries in computation. After the computation the two symmetric differences appearing in the result are recast back to sums for their intuitive appeal.

We will now turn our attention to some of the constructions available for XOR hierarchies.

Complex hierarchies may be obtained from the simpler ones. This is especially easy to do with XOR hierarchies. Let $\boldsymbol{a}_i$, $i = 1, 2, \ldots, n$ be a finite family of pairwise discrete shapes and let $X(\boldsymbol{a}_i)$ be a family of XOR hierarchies that analyze these shapes. The union of family $X(\boldsymbol{a}_i)$ augmented by the sum of family a_i is a complex XOR hierarchy that analyzes the latter sum.

$$X\left(\textstyle\sum_{i=1,\ldots,n} \boldsymbol{a}_i\right) = \left\{\textstyle\sum_{i=1,\ldots,n} \boldsymbol{a}_i\right\} \cup \bigcup_{i=1,\ldots,n} X(\boldsymbol{a}_i). \qquad (1)$$

Formula (1) may be used recursively to build complex hierarchies from the simpler ones. This is similar to what we do when we construct hierarchical file structures on our computers by dragging already populated file folders into an empty file folder.

The construction of hierarchies may also go in the other direction and derive smaller hierarchies from the bigger ones. The simplest of such constructions is based on the fact that all the elements that are parts of any reducible element of a hierarchy form a hierarchy themselves. This hierarchy is a subhierarchy of the original hierarchy.

If, for example, the process of assembling an object a is described by hierarchy $X(\boldsymbol{a})$, then the same process for a component $\boldsymbol{b} \leq \boldsymbol{a}, \boldsymbol{b} \in X(\boldsymbol{a})$ is described by the subhierarchy $X(\boldsymbol{b}) \subseteq X(\boldsymbol{a})$ generated by $\boldsymbol{b}$ and defined by $X(\boldsymbol{b}) = \{\boldsymbol{x} \leq \boldsymbol{b} \mid \boldsymbol{x} \in X(\boldsymbol{a})\}$. Using this construction the process of manufacturing an object may be divided into separate processes of manufacturing its components that then may be done concurrently.

The construction above works well for the components of an object. If, in contrast, the goal is to manufacture a new object that is a part, but not a component of the original object, then subhierarchies do not work. There is another possibility that relies on the *relativization* of the structure of an analyzed shape to one of its parts.

Definition 5. If $\boldsymbol{a}$ is a shape, A its decomposition, and $\boldsymbol{b}$ its part then $\boldsymbol{b}$ is implicitly structured by decomposition $\{\boldsymbol{b}\} \cdot A = \{\boldsymbol{b} \cdot \boldsymbol{x} \mid \boldsymbol{x} \in A\}$, which is the relativization of the structure of $\boldsymbol{a}$ to $\boldsymbol{b}$ or the relativization of A to $\boldsymbol{b}$ denoted by $A/\boldsymbol{b}$ (Krstic, 2005). ■

For example, the hierarchy in Figure 8d is a decomposition of the shape (a) that has shape (b) as a part. Decomposition in Figure 8e is the relativization of the original hierarchy to shape (c). Note that the relativization is also a hierarchy, but this may not always be the case. There are summable or discrete hierarchies with relativizations that fail to be hierarchies. In contrast, XOR hierarchies behave in accordance with the following result.

Proposition 5. *The structure of XOR hierarchies is preserved under relativization (Krstic, 1996).* ■

Proposition 5 describes yet another nice property of XOR hierarchies that may be used to facilitate a transformation of one hierarchy into another. If some facilities are used to assemble an object according to some XOR hierarchical decomposition of the object, then the same facilities may be used to assemble a new object, which is a part of the original object, in accordance with the relativization of the original hierarchy to this part.

This may be carried a step further to allow for the changing of components of the original hierarchy. The hierarchy may be changed to recognize a shape that is not its element, but is a part of the shape the hierarchy analyzes. Let $X(\boldsymbol{a})$ be an XOR hierarchy and shape $\boldsymbol{b} \notin X(\boldsymbol{a})$ be a part of shape $\boldsymbol{a}$. Relativization $X(\boldsymbol{a})/\boldsymbol{b}$ is an XOR hierarchy and so is $X(\boldsymbol{a})/(\boldsymbol{a} - \boldsymbol{b})$. Because the sum of $\boldsymbol{b}$ and $\boldsymbol{a} - \boldsymbol{b}$ is $\boldsymbol{a}$, identity (1)

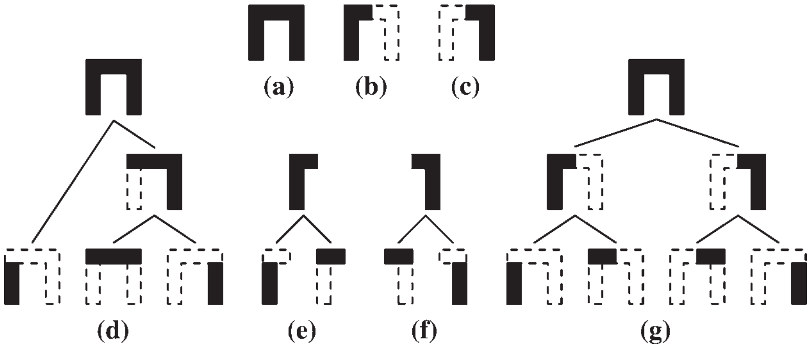

Fig. 8. The hierarchy construction: (a) shape and (b, c) its parts and, hierarchy (d) of (a) is changed by combining hierarchy (e), which is the relativization of (d) to (b), and hierarchy (f), which is the relativization of (d) to (c), to produce hierarchy (g).

https://doi.org/10.1017/S0890060409990102 Published online by Cambridge University Press

may be used to combine $X(\boldsymbol{a})/\boldsymbol{b}$ and $X(\boldsymbol{a})/(\boldsymbol{a} - \boldsymbol{b})$ in a new hierarchy $X'(\boldsymbol{a})$, or

$$X'(\boldsymbol{a}) = \{\boldsymbol{a}\} \cup X(\boldsymbol{a})/\boldsymbol{b} \cup X(\boldsymbol{a})/(\boldsymbol{a} - \boldsymbol{b}). \qquad (2)$$

For example, in Figure 8 hierarchy (d) of shape (a) is changed by taking hierarchy (e), which is the relativization of hierarchy (d) to shape (b), and hierarchy (f), which is the relativization of hierarchy (d) to shape (c), and combining them in accordance with (1) to get a new hierarchy (g).

Construction (2) allows for the components $\boldsymbol{b}$ and $\boldsymbol{a} - \boldsymbol{b}$ to be assembled via the same facilities that are used to assemble $\boldsymbol{a}$ according to $X(\boldsymbol{a})$. In contrast, assembling $\boldsymbol{a}$ from $\boldsymbol{b}$ and $\boldsymbol{a} - \boldsymbol{b}$ according to $X'(\boldsymbol{a})$ requires new facilities.

Formula (2) may be used recursively to gradually change a hierarchy by adding new parts either to the original hierarchy or to any of its subhierarchies.

3. TOPOLOGIES

Hierarchies are handy when conceiving the materialization of an already completed design, but may prove inadequate for the design process itself. Conjunctive combinations of entities are important in design especially in the early stages of the process. They yield new entities that have new properties not necessarily recognized by the original entities. This supports the discovery, which is an indispensable ingredient of the creative phases of the design process. Hierarchies do not support conjunction and a structure different from the hierarchical is to be used in the design process. The properties of such a structure will be given first informally and then formally in the remainder of the paper.

3.1. Definition

Consider the hierarchy in Figure 4 as a decomposition of shape (a) defined in $\boldsymbol{U}_{22}$. The latter is a shape algebra that manipulates planar segments in a plane. Note that Figure 4 also contains lines, dashed lines, and labels. Algebras for these would be needed as well, however, for our argument $\boldsymbol{U}_{22}$ alone will do.

Shape (a), which is analyzed by the decomposition, is its member, but also is the sum of the other members, shapes (b) through (k). Therefore, the decomposition represents a set of spaces of a house together with their sum, which is the plan of the house. Some other sums of interest are also included, like shapes (b) and (c) that represent the public and private spaces of the house.

There are reasons for including even more sums. For example, one may consider a middle section of the plan, which has no access to the perimeter walls. The area is bounded by the grid lines 3, 5, b, and d as shown in Figure 9a. Any space in need of artificial lighting and ventilation is a part of that shape. Such are the kitchen, bathroom, and closet. Their sum in Figure 9b may be included (in the decomposition) as it conveniently enumerates all the spaces with such a need. This shape becomes the greatest element that is a part of the shape in Figure 9a, or the greatest lower bound of that shape. Any part of the design may become interesting at some point in the design process. For every such part the decomposition should contain the greatest lower bound. Consequently, all of the sums have to be included and the decomposition is closed under the sum of $\boldsymbol{U}_{22}$. The advantages of this structure are twofold.

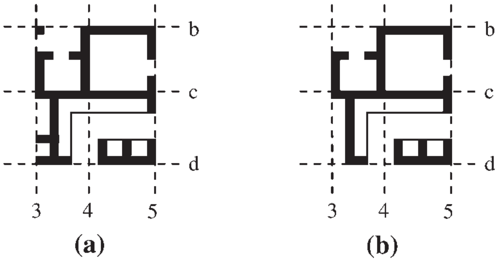

Fig. 9. The Ohlenbusch residence: (a) a middle section of the plan with no access to the perimeter walls and (b) the sum of the kitchen bathroom and closet.

First, it allows for the properties of any part of the design to be assigned to the appropriate elements of the decomposition. In the above example the properties of shape in Figure 9a are assigned to the kitchen, bathroom, and closet, labeling them as spaces in need of artificial ventilation and lighting.

Second, the structure comes in handy if a part of a design has to be defined in terms of properties that are recognized by the decomposition. For example, the shape in Figure 9a contains or has the properties of the kitchen, bathroom, and closet. This can be determined for every part of the design because there is a greatest lower bound for every such part.

To explore conjunctive combinations of elements of the decomposition we rely on the product operation from $\boldsymbol{U}_{22}$ algebra. For example in Figure 10, the product of shapes representing the kitchen (c) and bathroom (b) yields the shape (e), which represents the partitioning wall between the two spaces. Likewise, the product of shapes representing the bathroom (b) and closet (a) yields the partitioning wall (d). Taking products of adjacent spaces exhaustively enumerates such walls. The decomposition becomes a system of spaces together with partitioning walls between them. Further, products of shapes representing walls are joints between the walls.

For example, in Figure 10 shape (f) is the product of shapes (d) and (e). The latter shapes represent walls that makes (f) the joint between the walls. The decomposition now becomes a system of spaces, partitioning walls, and joints between the walls. Its structure is that of an algebra with respect to the sum, and a partial algebra with respect to the product. The fact that it has products rules out the possibility of the structure being a hierarchy.

Further elaboration of the design may require some additional constrains on the structure of the decomposition.

Suppose that the door between the bedroom and garage is to be removed and replaced with a wall. The smallest element

https://doi.org/10.1017/S0890060409990102 Published online by Cambridge University Press

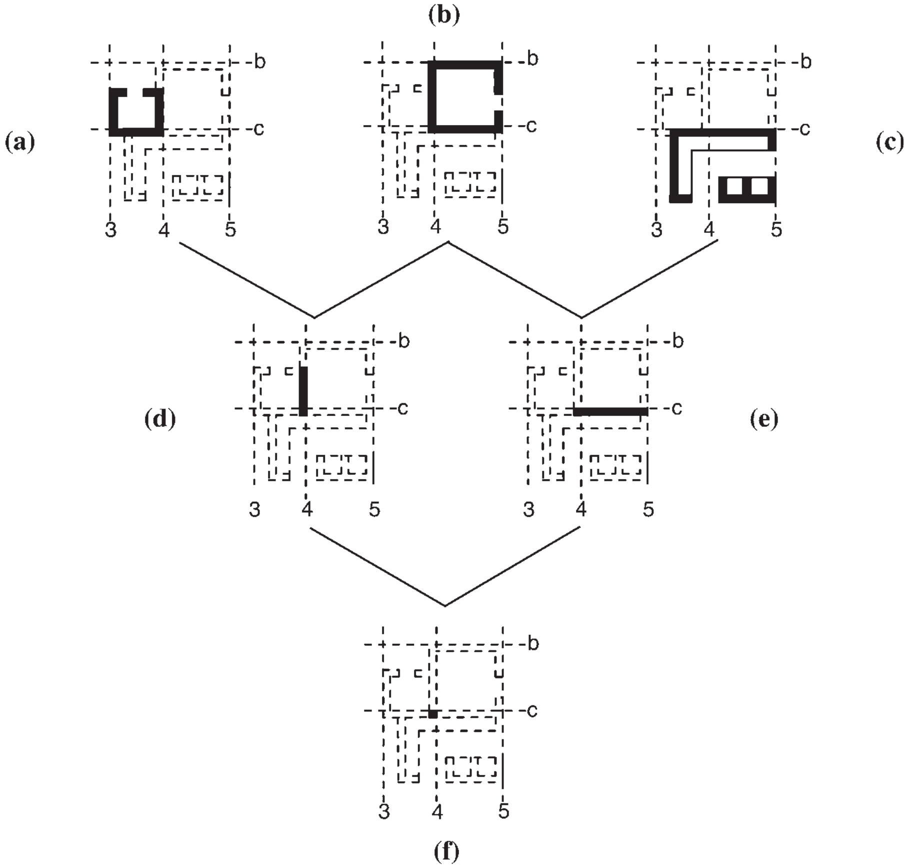

Fig. 10. The Ohlenbusch residence: (a) closet, (b) bathroom, and (c) kitchen; partitioning walls between (d) the bathroom and closet and (e) the kitchen and bathroom; and (f) the joint of walls (d) and (e).

of the decomposition altered by this action is the wall between the two spaces. It includes the door and is the least upper bound of the door with respect to this decomposition. To allow for a change of an arbitrary part of a design its decomposition should have the least upper bound for every such part. Consequently, the decomposition should be closed under products, which elevates its structure to an algebra closed under both sums and products.

Every part of the design is represented by the two distinguished elements of the decomposition. These are the least upper bound of the part and its greatest lower bound. The part is bounded by the two and has at least the properties of the latter element and at most the properties of the former one. The unique representatives are provided regardless of the fact that there are infinitely many parts while the decomposition is only finite. However, the decomposition need not be small. In our example, it has over 2000 elements, and we may include some new elements like, say, complements of the elements we already have.

Complements deserve a closer look. As discussed elsewhere (Krstic, 2004, 2005) we perceive a shape as a whole, but we break it down into parts to understand or explain it. We assign properties to the parts and tend to see the properties of the whole as the sum of the properties of the parts. This makes finding the complement of a part an easy task: just take the shape that has all of the properties of the whole but none of the properties of the part. Unfortunately, shapes usually do not meet our Boolean expectations.

For example, the complement of the kitchen should, in a Boolean fashion, consist of all the spaces of the house but the kitchen. Algebra $\boldsymbol{U}_{22}$ has a difference operation to support the creation of complements. The shape in Figure 11a is the complement of the kitchen obtained by subtracting the latter from the plan. It contains the bedroom and stairway, which are the only spaces that have no common walls with the kitchen. This shape is clearly not the complement we expected. It lacks the garage, bathroom, corridor, and living room.

In another example, we may try to determine the façade walls of, say, the living room. These are partitioning the outside space from the living room and, following the logic of the construction of partitioning walls, may be determined as the product of the two spaces. The outside space should be the complement of the sum of the inside spaces. Unfortunately, this complement is the empty shape here. Our entity is not a system of inside spaces surrounded by the outside space, but a system of walls, and only in the decomposition certain combinations of walls emerge as spaces. Consequently, complement is of no help in specifying the outside space. The latter has to be defined in the same fashion as all other spaces, as a combination of walls (Fig. 11b). The product of the living room and the outside space is the shape in

https://doi.org/10.1017/S0890060409990102 Published online by Cambridge University Press

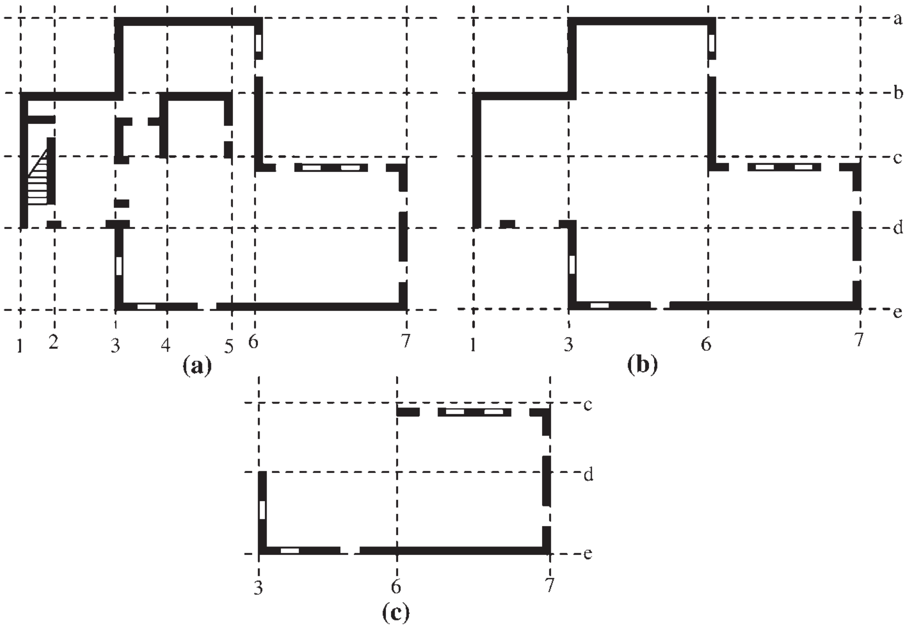

Fig. 11. The Ohlenbusch residence: (a) complement of the kitchen, (b) the outside space, and (c) the living room facade walls.

Figure 11c, which represents the façade walls of the living room.

Although the two examples show that complements are not essential in dealing with spatial entities there may be other compelling reasons for not using them. In an early, exploratory, stage of the design process designers may focus on developing certain parts of their designs while having only a vague notion of the other parts or the whole. Consequently, the complements may not be available at this stage. Without the complements we are left with an algebra which has the whole (the shape it analyzes) and zero (the empty shape), and is closed under + and · of $\boldsymbol{U}_{ij}$. This is the same structure as that of the closed (open) sets of a topological space that motivates introduction of the topologies for shapes.

Definition 6. Let a be a shape from $\boldsymbol{U}_{ij}$ algebra. A bounded decomposition of $\boldsymbol{a}$ is a *topology* on $\boldsymbol{a}$ or a *topological decomposition* of $\boldsymbol{a}$ if it is closed under + and · of $\boldsymbol{U}_{ij}$. A topological decomposition of $\boldsymbol{a}$ is denoted by $T(\boldsymbol{a})$. ■

3.2. Properties of topologies

Topologies for shapes were introduced by Stiny (1994) to study the continuity of shape computations. In contrast, we examine here how they may be used in place of shapes, as shape approximations.

A topological decomposition of a shape satisfies the unique representation requirement by providing for each part of the shape a pair of elements that bounds it. This is a far better representation than the one provided by hierarchical decompositions where elements, at best, top the shape parts. Topological decompositions of shapes have many interesting properties, but we will examine only those that pertain to topologies being used as shape approximations.

Because topology $T(\boldsymbol{a})$ is closed under + and · it has the least upper bound and the greatest lower bound for every part of $\boldsymbol{a}$. The two bounds define a topological closure Γ and a topological interior Ω operators on a set $\mathbf{B}(\boldsymbol{a})$ of all parts of $\boldsymbol{a}$ such that $T(\boldsymbol{a})$ is their image, or $T(\boldsymbol{a}) = \Gamma(\mathbf{B}(\boldsymbol{a})) = \Omega(\mathbf{B}(\boldsymbol{a}))$.

Stiny (1994) defines topology $T(\boldsymbol{a})$ as a set of closed elements of Boolean algebra $\mathbf{B}(\boldsymbol{a})$ that has been elevated to a closure algebra by the addition of a closure operator Γ. Stiny allows for topologies to be infinite, but requires for each part $\boldsymbol{x}$ of $\boldsymbol{a}$ an existence of a smallest element in $T(\boldsymbol{a})$ that contains $\boldsymbol{x}$ as a part. This requirement is weaker than the standard requirement for a topology to be closed under all products, finite and infinite. Because $\mathbf{B}(\boldsymbol{a})$ is not a complete Boolean algebra there is no guarantee that all of its infinite subsets have products, so the stronger requirement does not hold in infinite topologies for shapes. A topology that only satisfies the weaker requirement cannot be generated by a given subbase in a usual way. Because this construction plays an important role in applications, we will restrict ourselves to finite topologies only, in accordance with Definition 6 above.

For example, we constructed the topology of the Ohlenbusch residence from a subbase consisting of spaces in Figure 4d–k. We later modified the subbase by adding a shape representing the outside space (Fig. 11b). This resulted in a bigger topology that included the original one.

A topology generated by a subbase is the smallest among the topologies containing the subbase: all of the topologies containing the subbase contain this topology as well.

There is a simple procedure for generating a topology from a subbase (see, e.g., Vickers 1989, p. 32), but it may not be practical to keep all of the elements of a topology at all times. Even small designs may have big topologies like the topology in our example, which has over 2000 elements. It should be sufficient to keep the most important elements and construct

https://doi.org/10.1017/S0890060409990102 Published online by Cambridge University Press

the other elements as needed. In our example, the original spaces, which form the subbase, should be kept and we may also keep some other shapes like the walls and joints between the walls. We need to be able to construct the relevant elements to uniquely represent an arbitrary part of the analyzed shape. These are the closure and interior of the part, which could both be constructed directly from the subbase via the following procedure.

PROCEDURE 1. Let $\boldsymbol{a}$ be a shape, $\boldsymbol{x} \leq \boldsymbol{a}$ its part and B a subbase of a topology $T(\boldsymbol{a})$ on $\boldsymbol{a}$. Note that $\Sigma B = \boldsymbol{a}$, which renders B a decomposition of $\boldsymbol{a}$ and allows for the relativization of B to $\boldsymbol{x}$, or $B/\boldsymbol{x}$, to be constructed.

1. Construct set $B/\boldsymbol{x} = \{\boldsymbol{x}\} \cdot B$, which is a decomposition of $\boldsymbol{x}$ that recognizes all of the divisions of $\boldsymbol{a}$ imposed by B.
2. For every $\boldsymbol{y_i} \in B/\boldsymbol{x}$, where $i = 1, \ldots, n$ enumerates the elements of $B/\boldsymbol{x}$, construct two sets $C_i = \{z \in B \mid z \cdot \boldsymbol{x} \geq \boldsymbol{y_i}\}$ and $I_i = \{z \in B \mid z \cdot \boldsymbol{x} \leq \boldsymbol{y_i}\}$.
3. Construct a product ΠC_i for each C_i and a sum ΣI_i for each I_i. The product is the closure and the sum is the interior of $\boldsymbol{y_i}$ with respect to topology $T(\boldsymbol{a})$, or $\Gamma(\boldsymbol{y_i}) = \Pi C_i$ and $\Omega(\boldsymbol{y_i}) = \Sigma I_i$.
4. Finally, the sum of the closures above is the closure and the sum of the interiors above is the interior of $\boldsymbol{x}$ with respect to $T(\boldsymbol{a})$, or $\Gamma(\boldsymbol{x}) = \Sigma_{i=1,\ldots,n} \Gamma(\boldsymbol{y_i})$ and $\Omega(\boldsymbol{x}) = \Sigma_{i=1,\ldots,n} \Omega(\boldsymbol{y_i})$. ■

For example, in Figure 12 shape (a), defined in $\boldsymbol{U_{22}}$ algebra, is analyzed via topology (b) generated by subbase (c). Shape (d) is a part of (a) and its closure is shape (e). To construct (e) we take the relativization (f) of the subbase to shape (d) and construct the sets C_1, C_2, and C_3, which are a singleton (g), a set (c), and a singleton (i), respectively. Shapes (g), (h), and (i) are the products ΠC_1, ΠC_2, and ΠC_3, respectively, and their sum is the closure (e).

Note that Procedure 1 above provides for a direct construction of both $\Gamma(\boldsymbol{x})$ and $\Omega(\boldsymbol{x})$. We do not construct one and then use standard topological identities

$$\begin{aligned} \Omega_\Gamma(\boldsymbol{x}) &= \boldsymbol{a} - \Gamma(\boldsymbol{a} - \boldsymbol{x}), \\ \Gamma_\Omega(\boldsymbol{x}) &= \boldsymbol{a} - \Omega(\boldsymbol{a} - \boldsymbol{x}), \end{aligned} \tag{3}$$

to get the other. The identities above define operators that (in general) differ from Ω and Γ. If we consider $T(\boldsymbol{a})$ as a set of closed parts of $\boldsymbol{a}$ then it is the image of Γ. Its complement $T(\boldsymbol{a})' = \{\boldsymbol{a}\} - T(\boldsymbol{a})$ is the set of open parts of $\boldsymbol{a}$ and the image of Ω_Γ. Dually, $T(\boldsymbol{a})$ may be seen as the image of Ω, which makes it a set of open parts of $\boldsymbol{a}$. The related set of closed parts is then the image of Γ_Ω and the complement of $T(\boldsymbol{a})$. Operators Γ and Ω define two different topologies on $\mathbf{B}(\boldsymbol{a})$ except when $T(\boldsymbol{a})$ has the structure of a Boolean algebra. In that case, $\Gamma = \Gamma_\Omega$, $\Omega = \Omega_\Gamma$, and Γ and Ω belong to the same topology.

One of the reasons we dropped the complements was that they may not be available in the early stages of a design process. In contrast, complements may play a role later in the process when properties of design(s) are better understood. We may then use $T(\boldsymbol{a})$ and its complement $T(\boldsymbol{a})'$ concurrently to represent/bound the shape parts. This is done by paring any of the two closures with any of the two interiors so that any of the four combinations (Γ, Ω), (Γ, Ω_Γ), (Γ_Ω, Ω), or (Γ_Ω, Ω_Γ) may be chosen to best fit a particular part.

Another important use of the complements may arise in the final phases of the design process when designs are specified for production. Such a specification usually sports a kit of discrete parts from which to build a design as well as some details on how this should be done. This means that we need a discrete decomposition of a shape that can then be extended to a hierarchical decomposition to show how the original

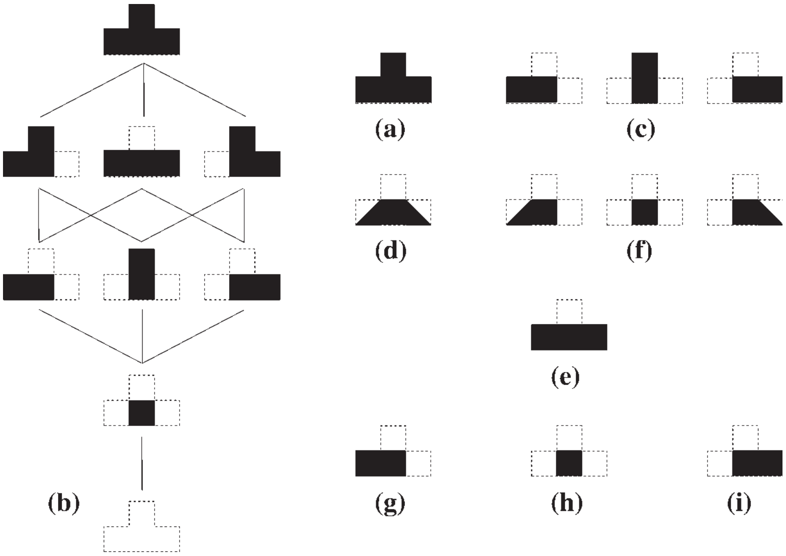

Fig. 12. Closure construction: (a) shape; (b) its topology with (c) the subbase; (d) a part and (e) its closure; (f) relativization of (c) to (d); (g, c, i) sets C_1, C_2, and C_3 with (g, h, i) their products, which sum to (e) the closure.

https://doi.org/10.1017/S0890060409990102 Published online by Cambridge University Press

discrete parts are put together to build the shape. There is no guarantee for topology $T(\boldsymbol{a})$ to be atomic so it may not contain a discrete decomposition of $\boldsymbol{a}$. To get one we need to extend $T(\boldsymbol{a})$ to a Boolean algebra in accordance with the following definition.

Definition 7. Let $T(\boldsymbol{a})$ be a topology on $\boldsymbol{a}$. The smallest decomposition of $\boldsymbol{a}$ that has the structure of a Boolean algebra and contains $T(\boldsymbol{a})$ is the *Boolean extension* of $T(\boldsymbol{a})$ denoted by $T_{\mathrm{Bool}}(\boldsymbol{a})$. ■

It is also a topological decomposition because it contains 0 and $\boldsymbol{a}$ and is closed under $\cdot$ and $+$ of $\boldsymbol{U_{ij}}$. In addition, $T_{\mathrm{Bool}}(\boldsymbol{a})$ is closed under the formation of complements, which is equivalent to being closed under $-$ of $\boldsymbol{U_{ij}}$. Because $T_{\mathrm{Bool}}(\boldsymbol{a})$ is finite Boolean algebra, it is atomic and its set of atoms A is a discrete decomposition of $\boldsymbol{a}$. Set A is the smallest, in terms of its cardinality, discrete decomposition of $\boldsymbol{a}$ such that all of the nonempty elements of $T(\boldsymbol{a})$ may be constructed as sums of elements of A. Moreover, all nonempty elements of $T_{\mathrm{Bool}}(\boldsymbol{a})$ are sums of atoms from A and they enumerate such sums. Because nonempty elements of XOR hierarchies are sums of their atoms (by Proposition 2 and Proposition 3), an arbitrary XOR hierarchical decomposition of $\boldsymbol{a}$, which has A as the set of atoms, can be embedded in $T_{\mathrm{Bool}}(\boldsymbol{a})$. This renders $T_{\mathrm{Bool}}(\boldsymbol{a})$ the least upper bound of all of such hierarchies. Note that, unlike Boolean algebras, hierarchies are not uniquely determined by their atoms so that there are many XOR hierarchies with A as the set of atoms.

As with $T(\boldsymbol{a})$, it is not practical to keep all of the elements of $T_{\mathrm{Bool}}(\boldsymbol{a})$ when we may only need some important ones and the ability to construct the set of atoms of $T_{\mathrm{Bool}}(\boldsymbol{a})$ as well as the closure and interior for any part of $\boldsymbol{a}$. A procedure for constructing the set of atoms of $T_{\mathrm{Bool}}(\boldsymbol{a})$ is as follows.

Procedure 2. Let $\boldsymbol{a}$, $T(\boldsymbol{a})$, $T_{\mathrm{Bool}}(\boldsymbol{a})$, and A be as above and let set B with n elements be a subbase of $T(\boldsymbol{a})$.

1. Construct $\wp(B) - \emptyset$ the set of all subsets of B with the exception of the empty set. This set has $2^n - 1$ elements.
2. For every $S_i \in \wp(B) - \emptyset$, where $i = 1, \ldots, 2^n - 1$, construct a set $D_i = B - S_i$.
3. For every i construct a shape $\boldsymbol{x_i} = \Pi S_i - \Sigma D_i$. This shape is either an atom of $T_{\mathrm{Bool}}(\boldsymbol{a})$ or an empty shape.[3]
4. Take $(\cup_{i=1,\ldots,2^n-1} \{\boldsymbol{x_i}\}) - \{0\}$, which is the set A of atoms of $T_{\mathrm{Bool}}(\boldsymbol{a})$. ■

For example, topology $T(\boldsymbol{a})$ in Figure 12b is extended to a Boolean algebra $T_{\mathrm{Bool}}(\boldsymbol{a})$ by including the shapes delineated by dashed lines. The set of atoms of the extension is given in Figure 13. Note that in this particular example the union of $T(\boldsymbol{a})$ and its complement $T(\boldsymbol{a})'$ is $T_{\mathrm{Bool}}(\boldsymbol{a})$. Usually the latter algebra is greater than this union.

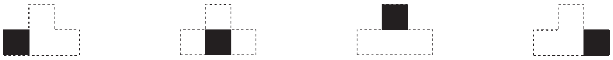

Fig. 13. The set of atoms of the Boolean extension of topology in Figure 12b.

Closure and interior of an arbitrary part $\boldsymbol{x}$ of shape $\boldsymbol{a}$ with respect to topology $T_{\mathrm{Bool}}(\boldsymbol{a})$ may be obtained in accordance with the following result.

Proposition 6. *Let a be a shape, x its part, $T(\boldsymbol{a})$ a topology on* $\boldsymbol{a}$, *Γ and Ω closure and interior operators from $T(\boldsymbol{a})$, and $T_{Bool}(\boldsymbol{a})$ the Boolean extension of $T(\boldsymbol{a})$. Closure and interior of* $\boldsymbol{x}$ *with respect to $T_{Bool}(\boldsymbol{a})$ are*

$$\Gamma_{\mathrm{Bool}}(\boldsymbol{x}) = \Gamma(\boldsymbol{x}) \cdot \Gamma_{\Omega}(\boldsymbol{x}) = \Gamma(\boldsymbol{x}) \cdot (\boldsymbol{a} - \Omega(a - \boldsymbol{x})),$$
$$\Omega_{\mathrm{Bool}}(\boldsymbol{x}) = \Omega(\boldsymbol{x}) + \Omega_{\Gamma}(\boldsymbol{x}) = \Omega(\boldsymbol{x}) + (\boldsymbol{a} - \Gamma(\boldsymbol{a} - x)), \qquad (4)$$

respectively. ■

Because $\boldsymbol{x}$ is a part of both $\Gamma(\boldsymbol{x})$ and $\Gamma_{\Omega}(\boldsymbol{x})$ it is a part of $\Gamma(\boldsymbol{x}) \cdot \Gamma_{\Omega}(\boldsymbol{x})$ so that the latter is an upper bound of $\boldsymbol{x}$ in $T_{\mathrm{Bool}}(\boldsymbol{a})$. We need to show that there is no such shape $\boldsymbol{y}$ that satisfies the following conditions $\boldsymbol{y} \in T_{\mathrm{Bool}}(\boldsymbol{a})$, $\boldsymbol{y} \geq \boldsymbol{x}$ and $\boldsymbol{y} < \Gamma(\boldsymbol{x}) \cdot \Gamma_{\Omega}(\boldsymbol{x})$ (‘) so that $\Gamma(\boldsymbol{x}) \cdot \Gamma_{\Omega}(\boldsymbol{x})$ is indeed the least upper bound. Because $T_{\mathrm{Bool}}(\boldsymbol{a})$ is closed under – and contains $T(\boldsymbol{a})$ it also contains $T(\boldsymbol{a})'$ so that either $\boldsymbol{y} \in T(\boldsymbol{a})$ or $\boldsymbol{y} \in T(\boldsymbol{a})'$ or y is a new element that does not belong to any of the topologies. If $\boldsymbol{y} \in T(\boldsymbol{a})$, then $\boldsymbol{y} \geq \Gamma(\boldsymbol{x})$ and if $\boldsymbol{y} \in T(\boldsymbol{a})'$, then $\boldsymbol{y} \geq \Gamma_{\Omega}(\boldsymbol{x})$ in any case $\boldsymbol{y} \geq \Gamma(\boldsymbol{x}) \cdot \Gamma_{\Omega}(\boldsymbol{x})$ so that $\boldsymbol{y}$ is not the least upper bound of $\boldsymbol{x}$. If $\boldsymbol{y}$ is a new element then it is a combination of two elements $\boldsymbol{u} \geq \boldsymbol{x}$ and $\boldsymbol{v} \geq \boldsymbol{x}$, from one or both of the topologies. The possible combinations are

a. $\boldsymbol{y} = \boldsymbol{u} + \boldsymbol{v}$, where $\boldsymbol{u} \in T(\boldsymbol{a})$, $\boldsymbol{v} \in T(\boldsymbol{a})'$,
b. $\boldsymbol{y} = \boldsymbol{u} \cdot \boldsymbol{v}$, where $\boldsymbol{u} \in T(\boldsymbol{a})$, $\boldsymbol{v} \in T(\boldsymbol{a})'$,
c. $\boldsymbol{y} = \boldsymbol{u} - \boldsymbol{v}$, where $\boldsymbol{u} \in T(\boldsymbol{a})$, $\boldsymbol{v} \in T(\boldsymbol{a})'$, and
d. $\boldsymbol{y} = \boldsymbol{u} - \boldsymbol{v}$, where $\boldsymbol{u}, \boldsymbol{v} \in T(\boldsymbol{a})$.

Other possible combinations are either symmetrical to the ones above or yield elements already in $T(\boldsymbol{a})$ or $T(\boldsymbol{a})'$. First, we show that there is no $\boldsymbol{y}$ obtained in combinations a to d, such that $\boldsymbol{y}$ satisfies (′). Combination a gives $\boldsymbol{y} \geq \boldsymbol{u} \geq \Gamma(\boldsymbol{x}) \geq \Gamma(\boldsymbol{x}) \cdot \Gamma_{\Omega}(\boldsymbol{x})$. In b case $\boldsymbol{u} \geq \Gamma(\boldsymbol{x})$ and $\boldsymbol{v} \geq \Gamma_{\Omega}(\boldsymbol{x})$ so that $\boldsymbol{y} = \boldsymbol{u} \cdot \boldsymbol{v} \geq \Gamma(\boldsymbol{x}) \cdot \Gamma_{\Omega}(\boldsymbol{x})$. Combination c yields $\boldsymbol{y} = \boldsymbol{u} - \boldsymbol{v} = \boldsymbol{u} \cdot (\boldsymbol{a} - \boldsymbol{v})$. Because both $\boldsymbol{u}$ and $(\boldsymbol{a} - \boldsymbol{v})$ belong to $T(\boldsymbol{a})$, so does their product $\boldsymbol{y}$ and $\boldsymbol{y} \geq \Gamma(\boldsymbol{x}) \cdot \Gamma_{\Omega}(\boldsymbol{x})$, as shown earlier. Finally, case d is the same as b because $\boldsymbol{y} = \boldsymbol{u} - \boldsymbol{v} = \boldsymbol{u} \cdot (\boldsymbol{a} - \boldsymbol{v})$, where $\boldsymbol{u} \in T(\boldsymbol{a})$ and $(\boldsymbol{a} - \boldsymbol{v}) \in T(\boldsymbol{a})'$. This proves that $\Gamma_{\mathrm{Bool}}(\boldsymbol{x}) = \Gamma(\boldsymbol{x}) \cdot \Gamma_{\Omega}(\boldsymbol{x})$, which with (3) results in $\Gamma_{\mathrm{Bool}}(\boldsymbol{x}) = \Gamma(\boldsymbol{x}) \cdot (\boldsymbol{a} - \Omega(\boldsymbol{a} - \boldsymbol{x}))$. The proof of the second identity of the proposition is a dual of the proof above.

Both Ω_{Bool} and Γ_{Bool} are expressed, in Proposition 6, in terms of Ω and Γ. Because the latter operators may be con-

[3] It is clearly so, because the product of $\boldsymbol{x_i}$ and any shape generated by S_i is $\boldsymbol{x_i}$ and the product of $\boldsymbol{x_i}$ and any shape generated by D_i is 0.

https://doi.org/10.1017/S0890060409990102 Published online by Cambridge University Press

structed from the subbase B via Procedure 1 construction of Ω_{Bool} and Γ_{Bool} is feasible.

We conclude this section with some counting to find out how big our structures may get. The size of any finite Boolean algebra including $T_{\text{Bool}}(\mathbf{a})$ is 2^k, where k is the number of atoms of the algebra. The latter number for $T_{\text{Bool}}(\mathbf{a})$ depends largely on the subbase B of $T(\mathbf{a})$, in accordance with Procedure 2 above. The size n of B matters, but also do the relations among its elements. In the maximum case, the only relation between the elements of B is $\Sigma B = \mathbf{a}$. Each shape $\mathbf{x_i}$, from Procedure 2, then yields a different atom so that the number of atoms of $T_{\text{Bool}}(\mathbf{a})$ is $k = 2^n - 1$. In the minimum case, the only elements of $T_{\text{Bool}}(\mathbf{a})$ are these of B, or $T_{\text{Bool}}(\mathbf{a}) = B$. Consequently, $n = 2^k$, so that the number of atoms is $k = \log_2 n$. We mentioned earlier that discrete (and by extension XOR) hierarchies with k atoms may range in sizes between $k + 1$ and $2k - 1$. This means that the simplest hierarchies embedded in $T_{\text{Bool}}(a)$ will have between $\log_2 n + 1$ and $2 \log_2 n - 1$ elements, whereas the most complicated ones will have between 2^n and $2^{n+1} - 3$ elements.

For example, in Figure 12 topology (b) generated by a three-element subbase (c) has the Boolean extension with four atoms depicted in Figure 13. This is less than the maximum seven because there are more relations than $\Sigma B = \mathbf{a}$ among the shapes of the subbase. In particular, products of different shapes of the subbase are equal. Similarly, the Ohlenbusch residence topology has been constructed from a subbase consisting of eight spaces, as shown in Figure 4d through 4k. The Boolean extension of such a topology may yield a maximum of 255 atoms. In contrast, we have here only about one-tenth of that number because of the numerous relations among the elements of the subbase.

3.3. Parts and elements

Because shape parts are represented via elements of topologies, the relation between the parts and elements is crucial to the understanding of how (well) shapes are approximated with the topologies.

For example, if some part $\mathbf{x}$ of shape $\mathbf{a}$ is examined so that the structure imposed on $\mathbf{a}$ by topology $T(\mathbf{a})$ is respected, then it is safe to say that $\mathbf{x}$ is $\Omega(\mathbf{x})$. Part $\mathbf{x}$ has the properties of $\Omega(\mathbf{x})$ and there is no other shape like this one in $T(\mathbf{a})$ that is greater than $\Omega(\mathbf{x})$. The part may have some other properties as well, but those are not recognized by $T(\mathbf{a})$. If, in contrast, the place of $\mathbf{x}$ in the structure of $T(\mathbf{a})$ has to be determined, then $\mathbf{x}$ is $\Gamma(\mathbf{x})$. Closure $\Gamma(\mathbf{x})$ accounts for all of the properties of $\mathbf{x}$ and is the least such element of $T(\mathbf{a})$. The closure may have some properties that $\mathbf{x}$ lacks, but $T(\mathbf{a})$ is oblivious to that.

For an arbitrary part $\mathbf{x}$ of shape $\mathbf{a}$ represented by $\Omega(\mathbf{x})$ and $\Gamma(\mathbf{x})$ in $T(\mathbf{a})$ there may be (infinitely) many parts of $\mathbf{a}$ with the same representation in $T(\mathbf{a})$. This justifies the introduction of an equivalence relation $\equiv$ such that $\mathbf{x} \equiv \mathbf{y}$ if and only if $\Omega(\mathbf{x}) = \Omega(\mathbf{y})$ and $\Gamma(\mathbf{x}) = \Gamma(\mathbf{y})$, where $\mathbf{x}, \mathbf{y} \leq \mathbf{a}$. If $\mathbf{x}$ is an element of $T(\mathbf{a})$, then $\Omega(\mathbf{x}) = \Gamma(\mathbf{x}) = \mathbf{x}$ and equivalence class $[\mathbf{x}]_{\equiv}$ has only one element $\mathbf{x}$. In contrast, when $\mathbf{x}$ is not an element of $T(\mathbf{a})$ class $[\mathbf{x}]_{\equiv}$ has infinitely many elements provided that $\mathbf{x}$ is not made of points only.

Number $n_{\equiv}$ of equivalence classes modulo $\equiv$ indicates how well $T(\mathbf{a})$ approximates $\mathbf{a}$. This number in the case of the smallest (trivial) topology $\{0, \mathbf{a}\}$ is 3. It appears that the trivial topology is not a bad shape approximation after all. It shows that $\mathbf{a}$ is nonempty, that it contains 0 and itself, and has proper parts as well. The following result provides some insights in the possible sizes of $n_{\equiv}$.

Proposition 7. *Let $T(\mathbf{a})$ be a topology on $\mathbf{a}$. For every comparable pair $\mathbf{x} \leq \mathbf{y}$ of elements of $T(\mathbf{a})$ there is a nonempty equivalence class of parts of $\mathbf{a}$ that are represented by $\mathbf{x}$, as their interior, and $\mathbf{y}$, as their closure, so that $n_{\equiv}$ equals the number of such pairs.* ■

In case $\mathbf{x} = \mathbf{y}$ there is a one-element equivalence class $[\mathbf{x}]_{\equiv}$. To show the same for $\mathbf{x} < \mathbf{y}$ it is enough to construct a shape $\mathbf{z}$ such that $\Omega(z) = \mathbf{x}$ and $\Gamma(z) = \mathbf{y}$. We start with $T_{\text{Bool}}(\mathbf{a})$, which is an atomic Boolean algebra with elements that are sums of atoms. Shape $\mathbf{y} - \mathbf{x}$ is an element of $T_{\text{Bool}}(\mathbf{a})$ and is the sum of some set of atoms A. We now construct a set B by taking a nonempty proper part from each of the elements of A, or $B = \{\mathbf{p} \mid \mathbf{p} < \mathbf{q}, \mathbf{p} > 0, \mathbf{q} \in A\}$. Shape $\mathbf{z} = \mathbf{x} + \Sigma B$ satisfies both $\Omega(z) = \mathbf{x}$ and $\Gamma(z) = \mathbf{y}$. The former because any shape $\mathbf{u} \in T(\mathbf{a})$, such that $\mathbf{x} < \mathbf{u} \leq \mathbf{y}$, contains at least one atom from A so that $\mathbf{z} \geq \mathbf{u}$ does not hold, and the latter because any shape $\mathbf{v} \in T(\mathbf{a})$, such that $\mathbf{x} \leq \mathbf{v} < \mathbf{y}$, misses at least one atom from A so that $\mathbf{z} \leq \mathbf{v}$ does not hold. Note that there are infinitely many choices for picking proper parts of atoms from A, when constructing B, so that there are infinitely many shapes in $[\mathbf{x}]_{\equiv}$, provided that $\mathbf{a}$ is not made of points.

In a topology that is a chain every pair of different elements is comparable so that for an n-element chain $n_{\equiv} = n\,(n + 1)/2$. This establishes the upper bound for $n_{\equiv}$. The lower bound is difficult to establish, but it is clearly greater than $3n - 3$ because for each $\mathbf{x} \in T(\mathbf{a})$ there is at least three classes, a one-element class $[\mathbf{x}]_{\equiv}$, a class defined by $\mathbf{x}$ and 0, and one defined by $\mathbf{a}$ and $\mathbf{x}$. Judging by the number of equivalence classes it seems that an n-element topology is much better shape approximation than an n-element hierarchy which only has n of such classes.

As with hierarchies, not all parts of $\mathbf{a}$ are equally well represented by topology $T(\mathbf{a})$. The elements of $T(\mathbf{a})$ are the parts that are best represented. They are represented by themselves. In contrast, parts that are both *boundary*, defined by $\Omega(\mathbf{x}) = 0$, and dense are the worst represented ones. Such parts have no recognizable properties despite the fact that only $\mathbf{a}$ is big enough to account for all of their properties. No matter how well a topology approximates a shape there will be infinitely many of such parts in accordance with the following corollary of Proposition 7.

Corollary 2. *Let $T(\mathbf{a})$ be a topology on $\mathbf{a}$ defined in $\mathbf{U_{ij}}$, where $i > 0$: $\mathbf{a}$ is not made of points. There are infinitely many parts of a that are both dense and boundary with respect to $T(\mathbf{a})$.* ■

In accordance with Proposition 7 there is a nonempty equivalence class defined by shapes 0 and $\boldsymbol{a}$, which are comparable. An element $\boldsymbol{x}$ of this class is both dense and boundary because $\Gamma(\boldsymbol{x}) = \boldsymbol{a}$ and $\Omega(\boldsymbol{x}) = 0$. There are infinitely many such parts in accordance with the last note of the proof of Proposition 7.

Although poorly represented, dense and boundary parts are well analyzed by the topology.

For example, the shape in Figure 14a is both dense and a boundary part of the shape in Figure 12a with respect to the topology in Figure 12b. The shape is, thus, poorly represented in the topology. In contrast, the topology in Figure 14b is the relativization of the topology in Figure 12b to the shape in Figure 14a and the two topologies are isomorphic. Consequently, the shape in Figure 14a and the shape in Figure 12a are analyzed by the original topology in the same way. Note that, not unlike XOR hierarchies, the structure of topologies is preserved under relativization.

We may assume that each part $\boldsymbol{x}$ of $\boldsymbol{a}$ is, de facto, approximated by the elements of $T(\boldsymbol{a})$ that represent it. Thus, we may calculate errors that such an approximation introduces. If $\boldsymbol{x}$ is approximated by $\Omega(\boldsymbol{x})$ the error is

$$\varepsilon_{\Omega}(\boldsymbol{x}) = \boldsymbol{x} - \Omega(\boldsymbol{x}), \tag{5}$$

whereas an approximation by $\Gamma(\boldsymbol{x})$ introduces the error

$$\varepsilon_{\Gamma}(\boldsymbol{x}) = \Gamma(\boldsymbol{x}) - \boldsymbol{x}. \tag{6}$$

Shapes that are elements of topologies are the most closely approximated ones, if $\boldsymbol{x} \in T(\boldsymbol{a})$ then $\varepsilon_{\Omega}(\boldsymbol{x}) = \varepsilon_{\Gamma}(\boldsymbol{x}) = 0$. In general, if $\boldsymbol{x}$ and $\boldsymbol{y}$ are parts of $\boldsymbol{a}$ and $\varepsilon_{\Omega}(\boldsymbol{x}) \leq \varepsilon_{\Omega}(\boldsymbol{y})$, then $\boldsymbol{x}$ is approximated by $\Omega(\boldsymbol{x})$ at least as well or better than $\boldsymbol{y}$.

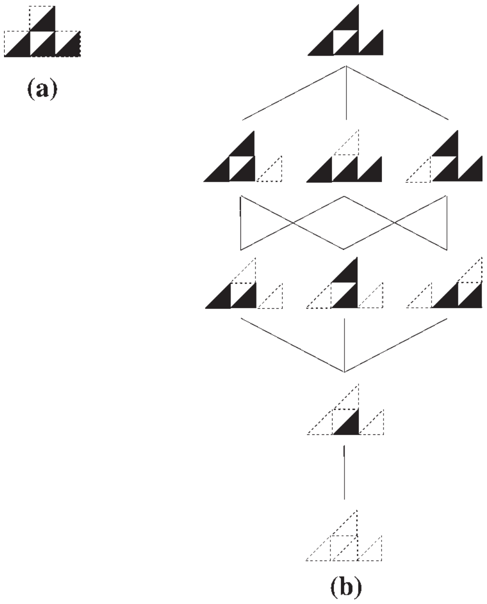

Fig. 14. (a) A part of the shape in Figure 12a, which is both dense and a boundary with respect to the topology in Figure 12b, and (b) the relativization of the topology to shape (a).

An interesting way of examining errors (5) and (6) is from the viewpoint of the topology itself. Because both $\varepsilon_{\Omega}(\boldsymbol{x})$ and $\varepsilon_{\Gamma}(\boldsymbol{x})$ are parts of $\boldsymbol{a}$ we may approximate them using different closures and interiors. A simple exercise in closure/interior arithmetic yields some interesting results.

For example, the Γ closure of $\varepsilon_{\Gamma}(\boldsymbol{x})$ is

$$\Gamma(\varepsilon_{\Gamma}(\boldsymbol{x})) = \Gamma(\Gamma(\boldsymbol{x}) - \boldsymbol{x}), \tag{7}$$

which is the smallest element of $T(\boldsymbol{a})$ that is greater than error $\varepsilon_{\Gamma}(\boldsymbol{x})$. The error cannot be greater than (7). Note that both error (6) and its closure (7) may be defined for XOR hierarchies because the latter are closed under Γ.

The Ω interior of $\varepsilon_{\Gamma}(\boldsymbol{x})$ is denoted by $\sigma(\boldsymbol{x})$ so that

$$\sigma(\boldsymbol{x}) = \Omega(\varepsilon_{\Gamma}(\boldsymbol{x})) = \Gamma(\boldsymbol{x}) \cdot \Omega(\boldsymbol{a} - \boldsymbol{x}). \tag{8}$$

The error cannot be smaller than $\sigma(\boldsymbol{x})$, which is the greatest element of $T(\boldsymbol{a})$ with a following property: is a part of $\Gamma(\boldsymbol{x})$ and has no common parts with $\boldsymbol{x}$. If $T(\boldsymbol{a})$ is a set of replaceable parts of some object and some part $\boldsymbol{x}$ breaks down, $\Gamma(\boldsymbol{x})$ will be replaced. Then again, because $\sigma(\boldsymbol{x})$ and $\boldsymbol{x}$ do not share parts the former shape may be recycled. If $\boldsymbol{a}$ is a car, $\Gamma(\boldsymbol{x})$ its alternator with a bad coil $\boldsymbol{x}$, then the repair shop uses $T(\boldsymbol{a})$ to pinpoint and replace the alternator. The old alternator goes to a remanufacturing facility as a "core." At the facility $\sigma(\boldsymbol{x})$ is used to find out which parts could be salvaged when rebuilding the alternator. Shape $\sigma(\boldsymbol{x})$ is the *core* of $\boldsymbol{x}$ with respect to $T(\boldsymbol{a})$ or the *reusable part* of $\Gamma(\boldsymbol{x})$.

As we have shown earlier, XOR hierarchies may be used the same way topologies are, as sets of replaceable parts for the maintenance purpose. In contrast, the concept of reusable part or core is foreign to hierarchies because their lack of the interior operator. This gives topologies an advantage over hierarchies in such applications. One can imagine a hierarchy, embedded in $T_{\text{Bool}}(\boldsymbol{a})$, that would recognize the core of the alternator, but such a hierarchy would also recognize its other components that will make it much bigger than $T(\boldsymbol{a})$. This hierarchy will do for assembling the car, but is too complicated for maintaining it.

Because closure Γ_{Ω} yields the complement of $T(\boldsymbol{a})$, we may take the closure of $\varepsilon_{\Gamma}(\boldsymbol{x})$ and then the complement of the result. This is another interesting shape and we will denote it by $\mu(\boldsymbol{x})$ so that

$$\mu(\boldsymbol{x}) = \boldsymbol{a} - \Gamma_{\Omega}(\varepsilon_{\Gamma}(\boldsymbol{x})) = \Omega(\boldsymbol{a} - \varepsilon_{\Gamma}(\boldsymbol{x})) = \Omega(\boldsymbol{x}) + \Omega(\boldsymbol{a} - \Gamma(\boldsymbol{x})). \tag{9}$$

This is the greatest element of $T(\boldsymbol{a})$ that does not have common parts with $\varepsilon_{\Gamma}(\boldsymbol{x})$. It has only those parts of $\Gamma(\boldsymbol{x})$ that are also parts of $\boldsymbol{x}$. If attention is paid to $\mu(\boldsymbol{x})$ and its parts only, then it does not matter whether $\boldsymbol{x}$ or its approximation $\Gamma(\boldsymbol{x})$ is used. From the point of view of $\mu(\boldsymbol{x})$ we make no error

https://doi.org/10.1017/S0890060409990102 Published online by Cambridge University Press

when approximating $\boldsymbol{x}$ is with $\Gamma(\boldsymbol{x})$. This shape becomes $\boldsymbol{a}$ if $\boldsymbol{x}$ is an element of $T(\boldsymbol{a})$.

Similarly, the complement of the Γ_Ω closure of $\varepsilon_\Omega(\boldsymbol{x})$ is

$$\boldsymbol{a} - \Gamma_\Omega(\varepsilon_\Omega(\boldsymbol{x})) = \Omega(\boldsymbol{a} - \varepsilon_\Omega(\boldsymbol{x})) = \Omega(\boldsymbol{x}) + \Omega(\boldsymbol{a} - \boldsymbol{x}). \qquad (10)$$

This is the greatest element of $T(\boldsymbol{a})$ that does not have common parts with $\varepsilon_\Omega(\boldsymbol{x})$. From the point of view of this shape, there is no error if $\boldsymbol{x}$ is approximated with $\Omega(\boldsymbol{x})$. This shape becomes $\boldsymbol{a}$ if $\boldsymbol{x}$ is an element of $T(\boldsymbol{a})$.

Because $\mu(\boldsymbol{x})$ is included in shape (10) their product is $\mu(\boldsymbol{x})$. The latter is the greatest element of $T(\boldsymbol{a})$ for which it does not matter whether x or any of its approximations is used. The greater $\mu(\boldsymbol{x})$ is, the better the topology approximates the part. We may use $\mu(\boldsymbol{x})$ as a measure of that approximation. Note that $\mu(\boldsymbol{x}) = \boldsymbol{a}$, if x is an element of $T(\boldsymbol{a})$ and $\mu(\boldsymbol{x}) = 0$, if $\boldsymbol{x}$ is both dense and boundary part of $\boldsymbol{a}$ with respect to $T(\boldsymbol{a})$. This measure cannot be defined for hierarchies, which makes it is hard to decide if a part is well approximated by a hierarchy.

The closure Γ of $\varepsilon_\Omega(\boldsymbol{x})$ is

$$\Gamma(\varepsilon_\Omega(\boldsymbol{x})) = \Gamma(\boldsymbol{x} - \Omega(\boldsymbol{x})). \qquad (11)$$

The error cannot be greater than this element of $T(\boldsymbol{a})$. In contrast, the error cannot be smaller than

$$\Omega(\varepsilon_\Omega(\boldsymbol{x})) = 0. \qquad (12)$$

Identity (12) renders $\varepsilon_\Omega(\boldsymbol{x})$ a boundary part of $\boldsymbol{a}$ with respect to $T(\boldsymbol{a})$. Similarly, $\Omega_\Gamma(\varepsilon_\Gamma(\boldsymbol{x})) = 0$ renders this error a boundary part of $\boldsymbol{a}$, but with respect to a different topology $T(\boldsymbol{a})'$.

The last of the possible combinations of the errors and operators is the complement of $\Omega_\Gamma(\varepsilon_\Omega(\boldsymbol{x}))$:

$$\boldsymbol{a} - \Omega_\Gamma(\varepsilon_\Omega(\boldsymbol{x})) = \Gamma(\boldsymbol{a} - \varepsilon_\Omega(\boldsymbol{x})) = \Gamma(\boldsymbol{a} - \boldsymbol{x}) + \Omega(\boldsymbol{x}). \qquad (13)$$

This shape, although an element of $T(\boldsymbol{a})$, is related to its complement $T(\boldsymbol{a})'$. It is the complement of the core of $(\boldsymbol{a} - \boldsymbol{x})$ with respect to $T(\boldsymbol{a})'$.

Finally, the two errors are bounded by the elements (12), (11), (8), and (7) of $T(\boldsymbol{a})$, so that

$$\begin{gathered} 0 \leq \varepsilon_\Omega(\boldsymbol{x}) \leq \Gamma(\boldsymbol{x} - \Omega(\boldsymbol{x})), \\ \Gamma(\boldsymbol{x}) \cdot \Omega(\boldsymbol{a} - \boldsymbol{x}) \leq \varepsilon_\Gamma(\boldsymbol{x}) \leq \Gamma(\Gamma(\boldsymbol{x}) - \boldsymbol{x}). \end{gathered} \qquad (14)$$

Those elements approximate (represent) errors $\varepsilon_\Gamma(\boldsymbol{x})$ and $\varepsilon_\Omega(\boldsymbol{x})$ in $T(\boldsymbol{a})$. The second of inequalities (14) may apply to XOR hierarchies, however, with a trivial lower bound.

$$0 \leq \varepsilon_\Gamma(\boldsymbol{x}) \leq \Gamma(\Gamma(\boldsymbol{x}) - \boldsymbol{x}). \qquad (15)$$

4. CONCLUSION

It has been shown earlier that bounded decompositions and their algebras behave the same way as shapes and shape algebras do (Krstic, 2004, 2005). The same holds for hierarchies and topologies for shapes, which are special kinds of bounded decompositions. What sets them apart are their respective algebraic structures. The latter have many important properties to facilitate their application as shape approximations.

Both hierarchies and topologies are expressive. Chief among their properties is their ability to uniquely represent each of the infinitely many parts of the shape they analyze.

Topologies are supportive. They provide for both conjunctive and disjunctive combinations of shapes. This yields new shapes to facilitate the discovery, which is important in the early creative stages of design. Hierarchies lack the disjunction so their ability to produce new shapes is greatly reduced rendering them inadequate at this stage of design.

Topologies are imprecise. Another property, beneficial at this stage, is their relaxed treatment of complements. Unlike Boolean algebras where each element calls for its complement, topologies do not mandate them. This is an advantage when, early in the design process, designers do not know all the parts of their designs. In contrast, hierarchies are very precise, requiring for all the design parts to be known, which renders their application early in the design process difficult.

Both topologies and hierarchies are flexible. Topologies grow as designers add new shapes to their subbases. Every time this happens new shapes emerge in combinations with the existing ones. If new shapes are added by shape rules, in the framework of shape grammars, topologies assure that this proceeds in a continuous fashion (Stiny, 1994, 2006). Hierarchies grow as new components are added and combined with the old ones via constructions (1) and (2).

As mentioned earlier, hierarchies are precise, but topologies could also be precise. When in the later stages of the design process definitive answers are required, topologies could be extended to Boolean algebras to yield kits of parts from which to build the objects. Boolean extensions of topologies are the least upper bounds for all the hierarchies constructed with these kits of parts. The hierarchies describe all of the possible ways the parts could be put together to make the objects.

Topologies are logical and so are hierarchies. We demonstrated that one can reason about the shape parts by using interior/closure arithmetic. With topologies, one can take full advantage of that, whereas with hierarchies, the possibilities are limited because of the fact that only the closure operator is available.

Topologies are metaphysical. Their lack of complements has some philosophical implications for design. Most formal systems are grounded in Boolean logic where the principal of the exclusion of the third allows for entities to be replaced by their complements. One does not have to construct an entity to know its properties. He/she can infer that from the complement. Mathematicians have a choice of either constructing the entity or proving its complement impossible, whichever is easier. Designers do not have this choice; the complement of a bridge is the running river. Designers construct, they do not prove. The logic of design is not Boolean but topological (Brouwerian). Interestingly enough, at the very basis of mathematics, theorems with constructive proofs are the only ones valid across different set theories. Designers have it right.

https://doi.org/10.1017/S0890060409990102 Published online by Cambridge University Press

ACKNOWLEDGMENTS

Special thanks to the reviewers for many useful suggestions that greatly improved the clarity of the paper.

REFERENCES

Birkhoff, G. (1993). *Lattice Theory*. Providence, RI: American Mathematical Society.

Earl, C. (1997). Shape boundaries. *Environment and Planning B: Planning and Design 24*, 668–687.

Krstic, D. (1996). *Decompositions of shapes*. PhD Thesis. University of California, Los Angeles.

Krstic, D. (1999). Constructing algebras of design. *Environment and Planning B: Planning and Design 26*, 45–57.

Krstic, D. (2001). Algebras and grammars for shapes and their boundaries. *Environment and Planning B: Planning and Design 28*, 151–162.

Krstic, D. (2004). Computing with analyzed shapes. In *Design Computing and Cognition '04* (Gero, J.S., Ed.), pp. 397–416. Dordrecht: Kluwer Academic.

Krstic, D. (2005). Shape decompositions and their algebras. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 19*, 261–276.

Lee, Y.T., & Requicha, A.A.G. (1982). Algorithms for computing the volume and other integral properties of solids I: known methods and open issues. *Communications of the ACM 25*, 635–641.

Ojeda, O.R. (1997). *The New American House 2: Innovations in Residential Design, 30 Case Studies*. New York: Whitney Library of Design, an imprint of Watson–Guptill Publications.

Stiny, G. (1991). The algebras of design. *Research in Engineering Design 2*, 171–181.

Stiny, G. (1992). Weights. *Environment and Planning B: Planning and Design 19*, 413–430.

Stiny, G. (1994). Shape rules: closure, continuity, and emergence. *Environment and Planning B: Planning and Design 21*, S49–S78.

Stiny, G. (2006). *Shape: Talking About Seeing and Doing*. Cambridge, MA: MIT Press.

Timmer, H., & Stern, J. (1980). Computation of global geometric properties of solid objects. *Computer Aided Design 10*, 301–304.

Vickers, S. (1989). *Topology Via Logic*. New York: Cambridge University Press.

Djordje Krstic received BA and MA degrees from Belgrade University and a PhD from UCLA. He taught at Belgrade University and was Principal Software Engineer at Alcatel–Lucent and Director of Software Development at Signalife. Dr. Krstic has been doing research in the field of design computation since 1988. His main research interests are in shape grammars, shape decompositions, algebras of design, and space syntax.

https://doi.org/10.1017/S0890060409990102 Published online by Cambridge University Press