NATIONAL
DIGITAL TWIN
PROGRAMME

A Framework for Composition

A Step Towards a Foundation for Assembly

cdbb
Centre for Digital Built Britain

Contents

Preface

The Construction Innovation Hub's Platform Design Programme[1] is working in partnership with both government and industry to develop a platform construction system consisting of a standardised kit of parts to deliver social infrastructure buildings, such as schools, hospitals, and village halls. This document is another piece of work done by the National Digital Twin programme in collaboration with the Construction Innovation Hub towards developing the Information Management Framework that will form the foundation for sharing consistent data not only for the Platform Design Programme, but for use across government and industry.

Industries involved in the creation and management of built assets require effective, resilient, and secure data sharing and aggregation. Much of this information is needed throughout the life of the asset and needs to be shared with a number of organisations. This is critical not only for asset management, but to support the services provided by the asset, as well as other considerations such as measuring the accumulating carbon emissions in order that a net zero footprint can be achieved.

As a result, a formal mechanism to ensure that the right information can be made available at the right time, to the right people and that the quality of the information is known and understood, is required.

The Information Management Framework is such a mechanism, the technical part of which comprises three main elements:

- A Foundation Data Model
- A Reference Data Library, and
- An Integration Architecture.

The Foundation Data Model (or ontology) and Reference Data Library define a common structure and meaning for information that is shared between organisations within and across sectors and domains. Together, therefore, they enable the consistent sharing and integration of information. The Integration Architecture comprises a combination of technologies that enables this sharing of data between data sources and the systems that use them.

1 https://constructioninnovationhub.org.uk/platform-design-programme-defines-the-need/

Summary

Component breakdowns are a vital multi-purpose tool and hence ubiquitous across a range of disciplines. Information systems need to be capable of storing reasonably accurate representations of these breakdowns. Most current information systems have been designed around specific breakdowns, without considering their general underlying formal structure. This is understandable, given the focus on devising the breakdown and that there is not a readily available formal structure to build upon. We make a step towards providing this structure here.

At the core of the notion of a component breakdown is the component as an integral (dependent) part of the composite whole. This leads to a rich formal structure, one that requires careful consideration to capture well enough to support the range of specific breakdowns. If one is not sufficiently aware of this structure, it is difficult to determine what is required to produce a reasonably accurate representation – in particular, one that is sufficiently accurate to support interoperability.

In this report, enabled by the Construction Innovation Hub, we describe this rich formal structure and develop a framework for assessing how well a data model (or ontology) has captured the main elements of the structure. This will enable people to both assess existing models as well as design new models. As a separate exercise, as an illustration, we develop a data model that captures these elements.

Associated with the notion of component (as an integral, dependent part) is the notion of replaceable part (see Appendix A for more details). We do not characterise this here but will do so in a later report.

Introduction

Engineers (for example) use many types of breakdown. They breakdown a product (a composite whole) into the components needed to assemble it and list these in a Bill of Materials (BOM). They breakdown the work on a project (another composite whole) into manageable chunks associated with deliverables and list these in a Work Breakdown Structure (WBS). There are a range of benefits of this approach. Managing the whole through its components is often easier than managing the whole on its own. The components may be 'modules' reusable in different products and projects – the products (or projects) may be 'modules' reusable across a range of bigger products and projects.

It is not just engineers that use breakdowns, they are ubiquitous across a range of disciplines that deal with both the natural and artefactual (for more on the topic, see Appendix B which has a brief overview of literature – see also Appendix C which gives a sketch of the linguistic depth through the etymology).

For engineers (and others) to collect and share information about breakdowns in computer information systems, they need to fit their representations accurately into a data structure. Here we explore how the general formal structure of component breakdown can be characterised in terms of a composing operation. We use this to develop a framework for assessing how data (or ontologies) have captured the structure.

Focus on the abstract general component breakdown structure

It turns out that a range of component breakdown structures – and their associated composing operations – emerge in many domains. In general, if a domain has a kind of (flat) part structure then it also has the potential for a sibling (hierarchical) composition structure. There are two clear cases (illustrated in Figure 1) rooted in mereonomy (which deals with wholes and parts) and taxonomy. Firstly, the breakdown of a particular individual into a whole-part structure – where, for example, my car can be broken down into its body, chassis, engine, interior and wheels. Secondly, a super-set sub-set structure of classes in taxonomies – where, for example, engines are broken down by their energy sources into heat, non-thermal, electrical and physically powered engines.

Much of the interest in component breakdowns has focused on specific breakdowns. There is some, but not much, work that looks at the component breakdown structure as such, abstracting away from particular examples (for example (Bittner et al., 2001)). One cannot help thinking that there is a common, mistaken, assumption that this general structure is in some sense obvious, or easily recoverable, so not worth examining closely. We start here by showing that the underlying formal structure, needed for information systems' data structures is neither obvious nor easily recoverable – though it is based upon a simple base structure, a composing relation. Accordingly, in order to develop a clearer general picture, in this initial section, we focus on the abstract general component breakdown structure, for the most part ignoring which specific breakdown it is and even which domain it might be in.

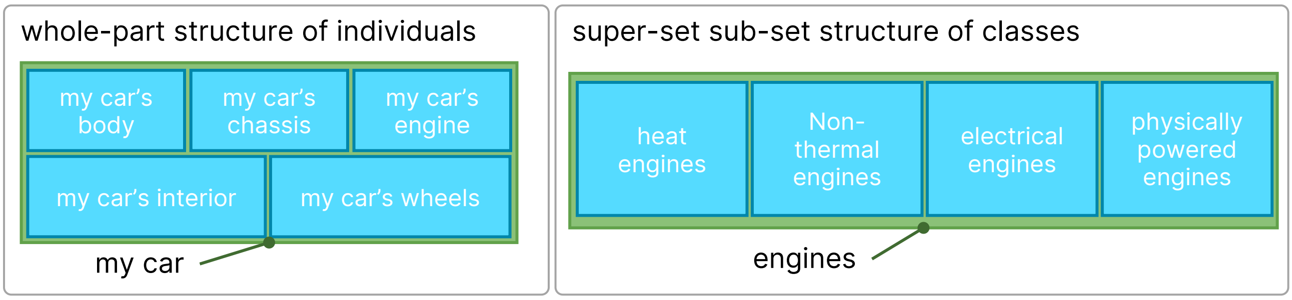


Figure 1: Two clear cases rooted in mereonomy and taxonomy

Structure of the paper

The body of the paper has two broad parts. In the first part, the general component breakdown structure is described in three sections. The first section describes the composing operation at the heart of the structure and how this is reified into a composition relation which we call an atomic composition. It notes the close connection between the atomic composition and whole-part relations into which it can be dissected. In the second section, it looks at constrained atomic compositions, where the composing objects are disjoint, what we call strict atomic compositions. It then looks at how these atomic compositions can be joined together to create hierarchies, which we call molecular compositions. In the third section, ways of relaxing the constraints are examined.

In the second broad part, there are also three sections. In the first, the analysis work in the first three sections is translated into an assessment framework. Then, briefly, further work is discussed and finally the report is summarised.

The composing operation

The rich formal structure of component breakdowns has been much commented on. We describe this hierarchical structure, and its associated flat part structure using a simple composing operation.

A key feature of breakdowns is a property of the composite wholes; completeness – in that they are composed of just their components, no more. This 'covering' composing property has been noticed and commented on since the Ancient Greeks (see Appendix D). It has an obvious practical side, if a component is missing from a BOM or a WBS, then the composite product or project is not whole, it is incomplete. In many engineering situations, it probably won't work or won't work properly. In a sense, studying breakdowns is studying wholes in relation to their parts and so involves a kind of 'holology' (or 'whololgy') – see (Cotnoir et al., forthcoming, p. 17).

This property of completeness is missing from the individual whole-part relations in the associated flat structure. By itself, the individual relation carries no implication that it is linked to any particular complete whole – and so does not involve whology. Though, as we shall see, given a plurality of such relations (with the right background theory) one can construct a whole – and so the composition property for that whole.

A simple constructional process

One can devise a simple general constructional process to generate both the composition and its associated whole-part structures (Partridge et al., 2017), (Partridge et al., 2019), (Partridge et al., forthcoming). One starts with a plurality – technically, a simple collection with no order and no repetition – of objects in a domain. This is input as components into the constructional process which builds ('assembles') the composite. The left-hand side of Figure 2 shows this, using a funnel to symbolise the operation. The centre has a graph representing the result of the operation. The right-hand side shows the result of dissecting the composition into whole-part relations.

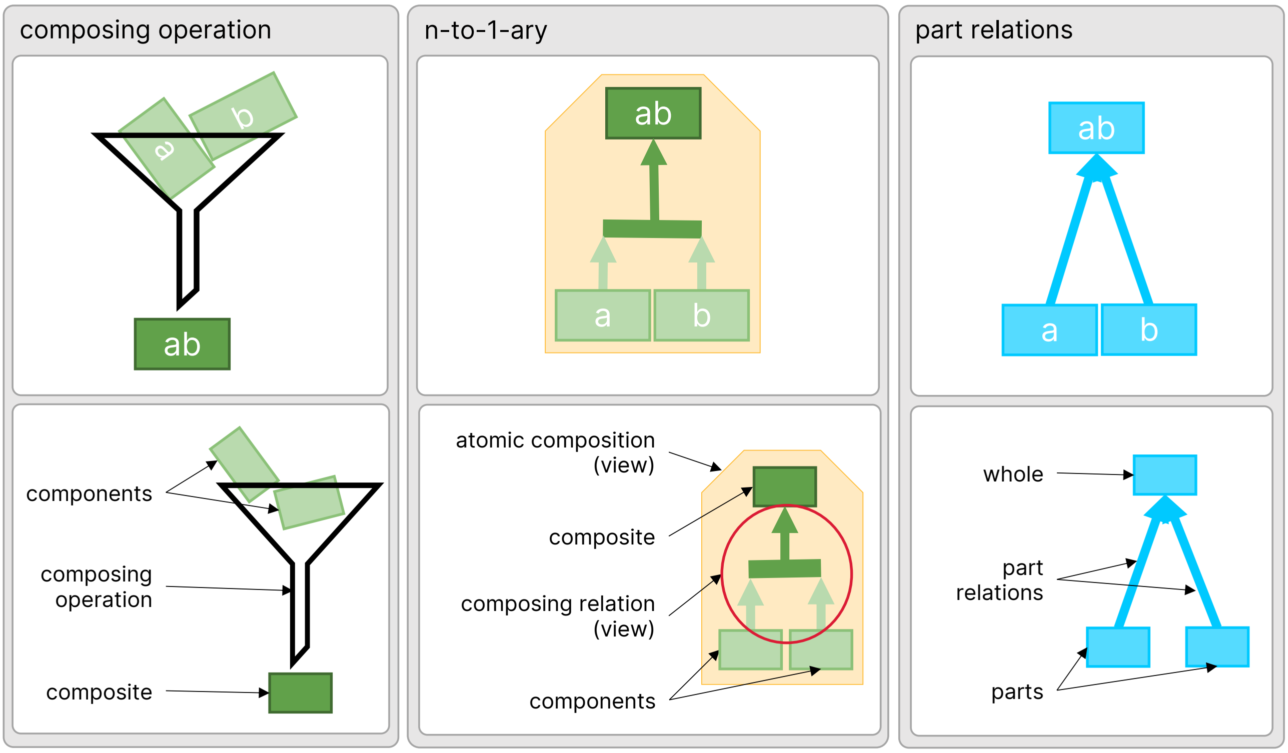


Figure 2: A simple constructional process

In this structure, components are (logically) assembled/composed into a single composite whole – so, in a sense, the composing operation (which we reify as a relation – or state of affairs – and call an 'atomic composition' – Figure 2 shows the relation and composition views) is an n-to-1-ary operation between the composite and the components (where the elements on the n-ary side of the relation are unique and unordered, in effect, a plurality).

Though there are multiple possible approaches to characterising composition, for expository simplicity, here we assume an underlying theory of parthood, from which composition emerges. Our favoured theory is classical mereology (to be found in many places, including (Cotnoir et al., forthcoming)). In this approach, the fusion operation is the engine for composition. The composition operation builds the composite as the fusion of the components. An alternative constructional approach might take the composition operation – and so composition – as primitive and derive the part relations. A transparent account of composition should make clear what approach it takes.

In practical terms, it makes little sense to physically assemble less than two components – to assemble a single, or (indeed) no, component. But, for overall symmetry and consistency one might want to allow these last two cases – visualised in Figure 3. If one is allowed an assembly with no components, then the result would be the null component (see (Fine, 2010)). If one allowed a single component, then as the composite would be the same single component, one would be allowing things to be improper components of themselves. A transparent account of composition should make clear where its structure stands on these two points. Going forward, our default assumption will be that we are dealing with 'proper' components, so each composition will have multiple components and there will not be a null component. Alternative approaches may decide to make different choices.

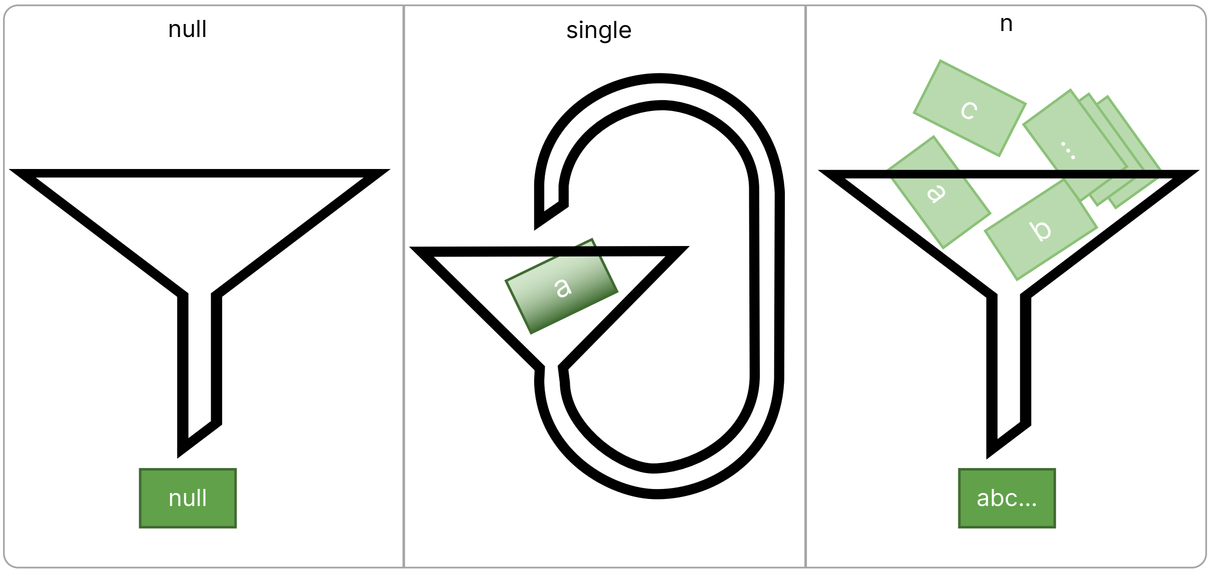


Figure 3: Formal overall symmetry and consistency cases

Where it makes sense, we include the formalisation in the body of the text, highlighted in grey – as below. We collect all the formalisation into one place, for ease of reference, in Appendix E.

> We make the fundamental propositions a little more rigorous by formalising them (in first-order logic). Our aim is not to provide a complete theory, only to delineate for the sake of clarity. We introduce the vocabulary we need, or just find useful, in formulating these propositions as we go. We make no attempt to provide a complete or minimal formalisation and we leave open whether it could be replaced or condensed by an alternative axiomatisation.
>
> Let us assume that:
>
> AtomicComposition(x) means x is an atomic composition.
>
> isAtomicComponentOf(x,y) means x is an atomic component of y, where y is an atomic composition.
>
> isAtomicCompositeOf(x,y) means x is an atomic composite of y, where y is an atomic composition.
>
> We can clarify that only atomic compositions have components and composites:

(P1) $\forall x\ \forall y\ ((isAtomicComponentOf(x,y) \lor isAtomicCompositeOf(x,y)) \rightarrow AtomicComposition(y))$

Furthermore, for a given composition, components and composites are distinct:

(P2) $\forall x\ \forall y\ (isAtomicComponentOf(x,y) \rightarrow \neg\ isAtomicCompositeOf(x,y))$

We refer to a standard account of the classical mereological fusion operator that allows us to denote the sum of all objects that satisfy a given property, ϕ, assuming the property has instances. This is standardly defined in terms of the Parthood relation. Under these assumptions, given a property, the fusion of the objects satisfying this property exists and is unique. This implies a sophisticated formal apparatus that we will use here for the convenience of presentation but recognise that we may need to consider an alternative treatment in a fuller formalisation.

$\sigma x \phi(x)$ denotes the fusion of the x's such that $\phi(x)$.

We can then use fusion to express the claim that the atomic composite of an atomic composition is the fusion of its atomic components:

(P3) $\forall x\ \forall y\ (AtomicComposition(x) \rightarrow (isAtomicCompositeOf(y,x) \equiv (y = \sigma z(isAtomicComponentOf(z,x)))))$

Notice that by construction of atomic compositions as (unique) fusions of atomic components, it is the case that every atomic composition has a unique atomic composite:

(P4) $\forall x\ \forall y\ \forall z\ ((isAtomicCompositeOf(x,y) \land isAtomicCompositeOf(z,y)) \rightarrow x=z)$ (Unicity of Atomic Composite)

To adopt the assumption that there are at least two components is to accept a proposition we will call the Multiplicity of Atomic Components principle.

(P5) $\forall x\ (AtomicComposition(x) \rightarrow \exists y\ \exists z\ (isAtomicComponentOf(y,x) \land isAtomicComponentOf(z,x)\ \neg\ (y=z)))$ (Multiplicity of Atomic Components)

Furthermore, as there seems to be no practical requirement for infinite composition from, for example, engineering practice, we shall assume here that the plurality of components is finite, which simplifies the formal structure. Again, for transparency, one should make clear where one's structure stands on this and the earlier overall symmetry and consistency points.

Let us call the results of a single composing operation, *atomic compositions* (later on, we join atomic compositions together to construct a hierarchy, which we call a *molecular composition*). The criterion of identity for atomic compositions is the same (plurality of) components. Two atomic compositions with the same components are the same composition.

We have already introduced the formalism for atomic compositions as fusions of components above. The criterion for identity of atomic compositions can now be formulated as follows:

(P6) $\forall x\ \forall y\ ((AtomicComposition(x) \land AtomicComposition(y) \land \forall z\ (isAtomicComponentOf(z,x) \equiv isAtomicComponentOf(z,y))) \rightarrow x=y)$ (Atomic Composition Criterion of Identity)

Associated part relation

Given an n-to-1-ary atomic composition relation, one can formally 'dissect' it into multiple binary dissected composing relations which correspond to part relations – as shown in Figure 4.

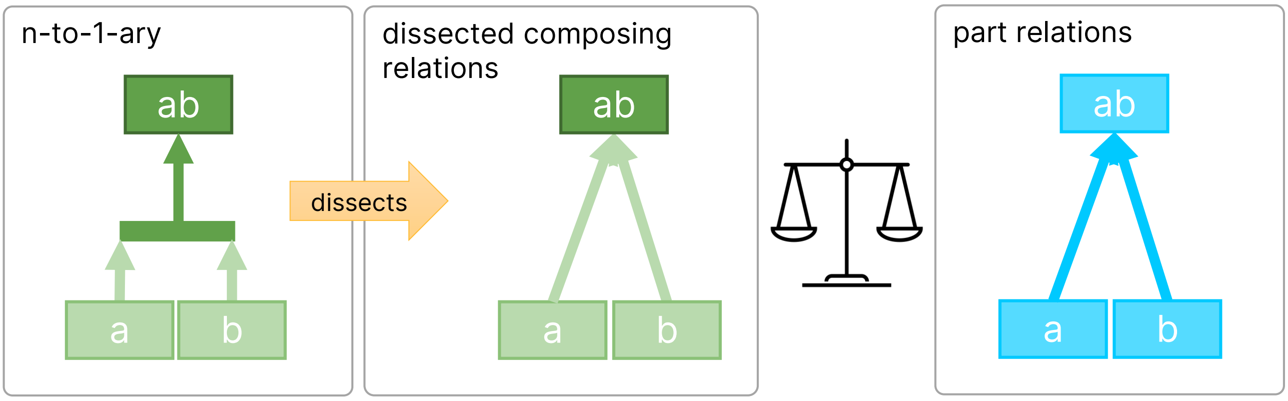


Figure 4: Harmonised dissected composing and part relations

In order to articulate this point, and others, we will use a formalisation in which we reify the particular binary relations between a part and a whole.

PartRelation(x) means that x is a particular part relation between objects. We use the predicates partInRelation and wholeInRelation to write 'partInRelation(a,b)' (respectively, 'wholeInRelation(c,b)') for a (respectively, c) is the part (respectively, the whole) in the part relation b.

Note that we can make the link to the parthood relation predicate as follows:

wholePart(x,y) means that y is a part of x.

(P7) ∀x ∀y ∀z ((PartRelation(x) ∧ partInRelation(y,x) ∧ wholeInRelation(z,x)) → wholePart(z,y))

dissectingPartInAtomicComposition(x,y) means x is an atomic composition for which y is one of the dissecting part relations, i.e., where y is a part relation instance and the whole in y is the composite of x and the part in y is an atomic component of x

Assuming an extensional criterion of identity, one can characterise these part relations – one part relation is identical with another if their whole and part projections are the same. They never appear twice in the same composition, though the same part relation can appear in multiple composition relations.

(P8) ∀x ∀y ∀u ∀v ((PartRelation(x) ∧ PartRelation(y) ∧ partInRelation(u,x) ∧ partInRelation(u,y) ∧ wholeInRelation(v,x) ∧ wholeInRelation(v,y)) → x=y) (Part Relation Criterion of Identity)

The part relations corresponding to the dissected composing relations uniquely define the composition, there is no other composition with exactly the same relations – in this sense, the part relations are a signature for the composition.

(P9) ∀x ∀y ((AtomicComposition(x) ∧ AtomicComposition(y) ∧ ∀z (dissectingPartInAtomicComposition(x,z) ≡ dissectingPartInAtomicComposition(y,z))) → x=y)

There naturally results in a requirement for a kind of ‘mereological harmony’ (used in a similar, but different, sense in (Uzquiano, 2011)), where the part and composition relations point in the same direction. Given this harmony, the part relations corresponding to the dissected composing relations are the domain’s part relations – if the dissected composing relation were the ‘wrong-way around’ – were not in harmony – then this would not happen – see Figure 5.

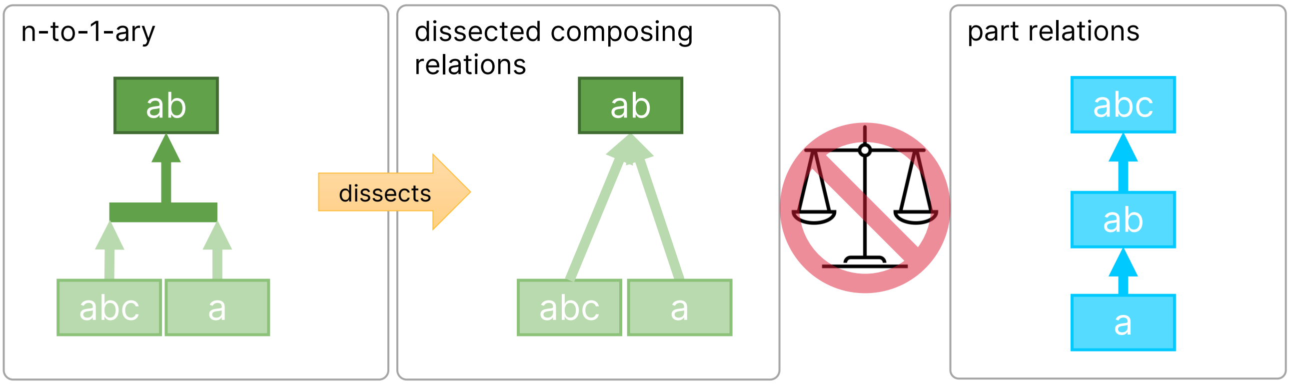


Figure 5: Disharmonised dissected composing and part relations

However, in the dissection one loses a clearly identified whole with integral parts. Just given a plurality of part relations corresponding to dissected composing relations for multiple compositions there is generally no way to recover the original composition or identify which elements are the original composites. Figure 6 illustrates this – the left-hand part relations can be used to reconstruct multiple compositions, without any indication of which one is intended. If one starts with a single atomic composition, dissect to its corresponding part relations, and then add another part relation – one can no longer recover exactly the original composition.

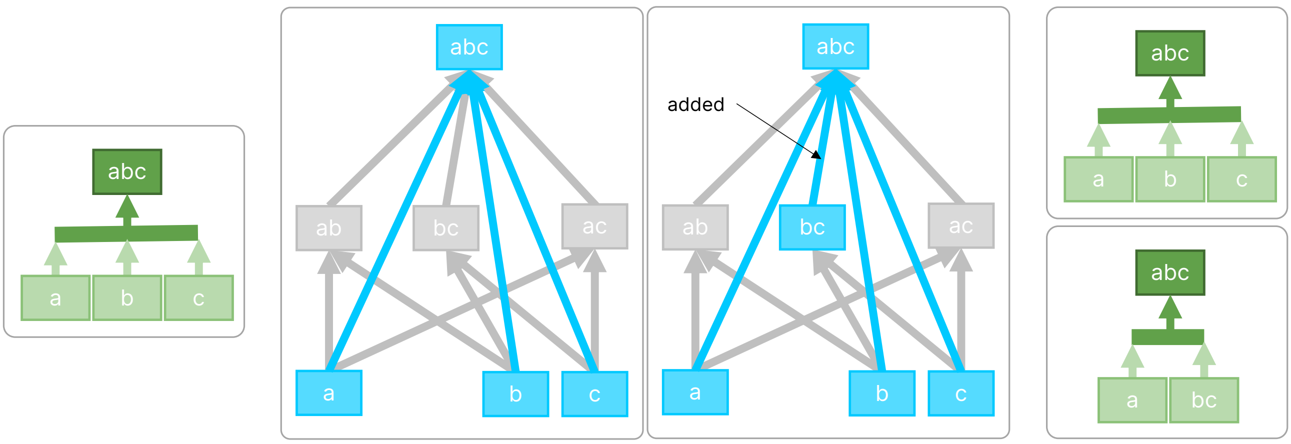


Figure 6: Example of a loss of composition information in part relations

Though, as the dissected parts are a signature for the composition, given the correct set of part relations one can recover the original composition. The set contains enough information to reconstruct the whole. More generally, there are sets of part relations that correspond one-to-one with the compositions.

> For atomic compositions, this one-to-one correspondence can be formalised as the equivalence between a component decomposition and the existence of a part relation for each atomic component of the atomic composition such that the atomic component is the part and the whole is the atomic composite.
>
> (P10) ∀x ∀y ∀z ((AtomicComposition(x) ∧ isAtomicComponentOf(y,x) ∧ isAtomicCompositeOf(z,x)) ≡ ∃r (PartRelation(r) ∧ dissectingPartInAtomicComposition(x,r) ∧ partInRelation(y,r) ∧ wholeInRelation(z,r)))
>
> Hence, every component of an atomic composition is a part of the composite of that composition (while, in general, the converse is not the case in the absence of a non-dissecting part relation).
>
> (P11) ∀x ∀y ∀z ((isAtomicComponentOf(y,x) ∧ isAtomicCompositeOf(z,x)) → wholePart(z,y))

A (very) simple example

It is perhaps easier to illustrate this with a simple example. Consider a case where there are two mereological atoms (objects with no parts). Assume we have a simple composing operation with no restrictions on what can be composed. As Figure 7 shows, we can fuse a and b to construct ab. This gives us a single composing relation which we can dissect into two (associated) binary whole-part relations.

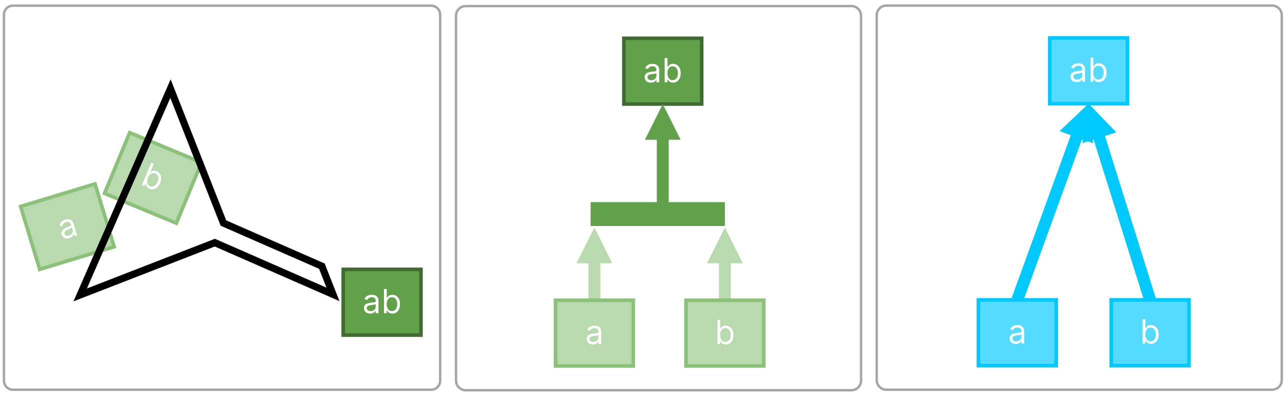


Figure 7: Simple case of two mereological atoms

This is too simple. Things get more interesting with three mereological atoms. We use fusion to construct all the possible wholes. After we have exhausted it (completed all the possible fusions) we end up with the part relations shown in Figure 8.

As noted earlier, compositions are atomic where there is only one composing operation and molecular when there is more than one – examples of these two kinds are shown in Figure 9. We consider atomic compositions initially.

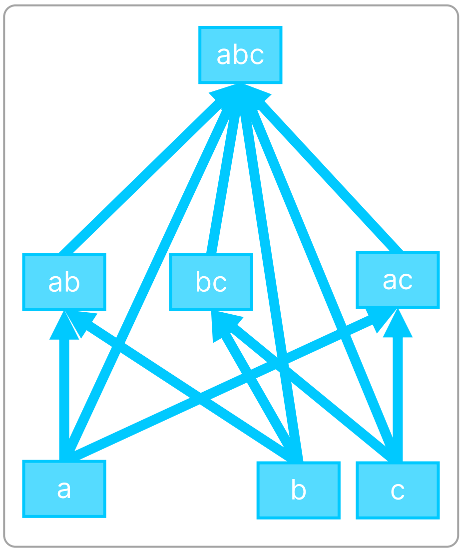


Figure 8: Simple case of three mereological atoms – all part relations

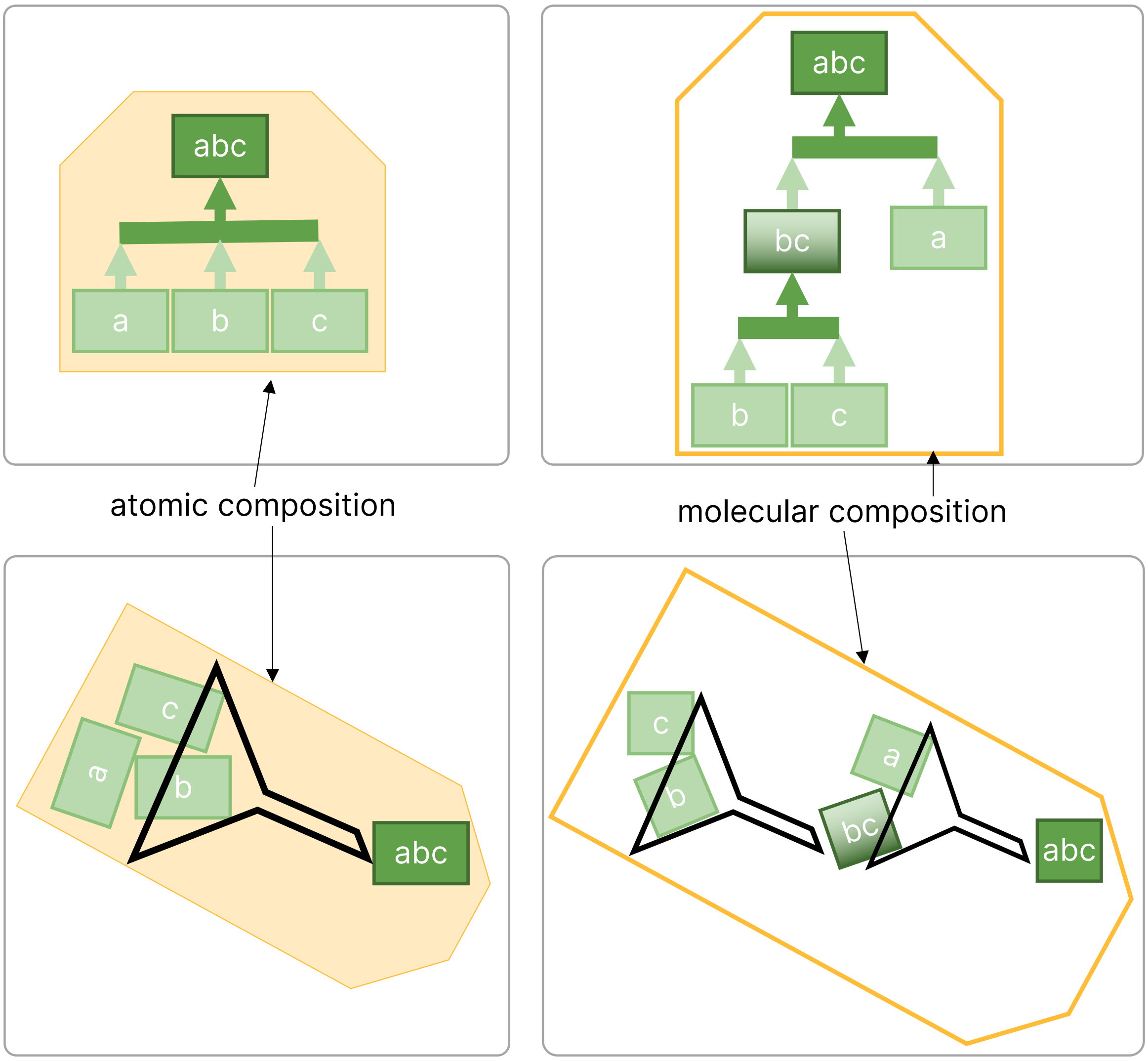


Figure 9: Examples of atomic and molecular composition

Let us call atomic compositions strict when the components of the composition are disjoint. As, in our example, the mereological atoms have no parts and so are, by definition, disjoint, we have a simple way, in this example, of determining disjointness. We look at their ultimate parts, the atoms from which they have been composed. If they share no atoms, then they are disjoint (this of course, only works where things are composed of mereological atoms).

> StrictAtomicComposition(x) means that x is a strict atomic composition.
>
> disjointFrom(x,y) means that x and y are mereologically disjoint, that is to say, according to standard mereological definitions, they do not share a part.
>
> (P12) ∀x (StrictAtomicComposition(x) ≡ (AtomicComposition(x) ∧ ∀y ∀z ((isAtomicComponentOf(y,x) ∧ isAtomicCompositeOf(z,x) ∧ ¬ (y=z)) → disjointFrom(z,y))))

In our example, there are four strict compositions with a as one of the components. The algorithm for finding this is: start with a in the part hierarchy on the left of Figure 10, find each whole of which it is a part then find all the disjoint parts of that whole. This clearly illustrates how a component can be composed in multiple ways.

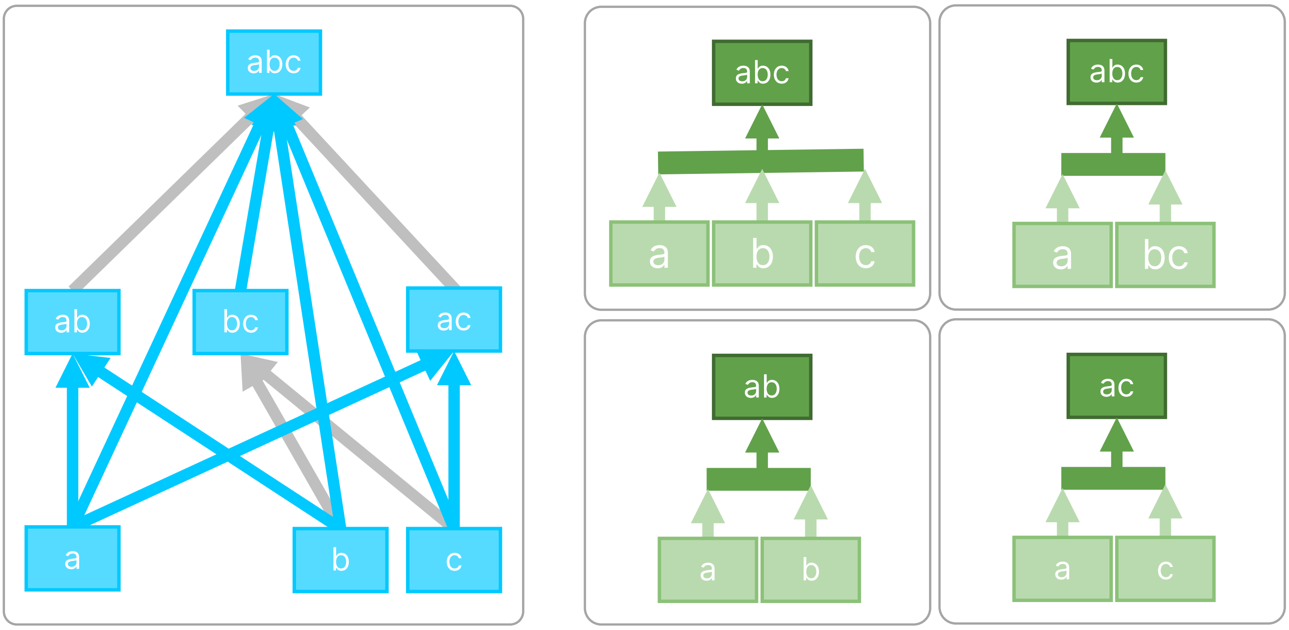


Figure 10: Example of four strict compositions with 'a' as one of the components

Furthermore, there are four strict compositions with abc as their composite – as illustrated in Figure 11. This clearly illustrates how a composite can be strictly decomposed in multiple ways (there are multiple other weaker decompositions).

One can view the last three atomic compositions in Figure 11 as the final stage in the construction history – where there are prior compositions. Figure 12 shows this for one of the cases. This shows the composition recapitulating the final stage of one of the composing histories of the composite.

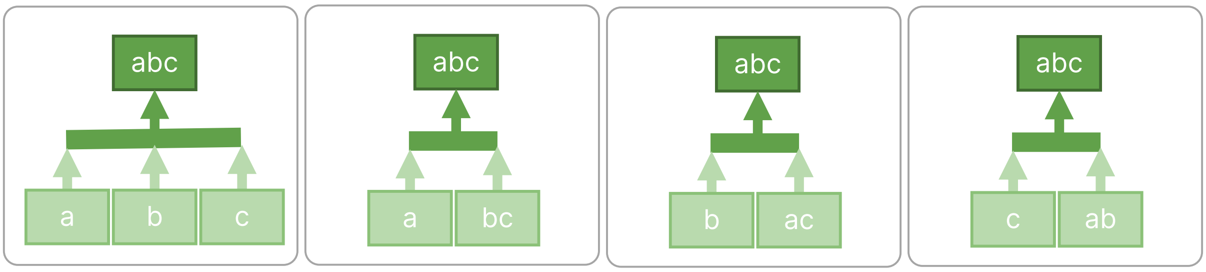


Figure 11: Example of four strict compositions with 'abc' as their composite

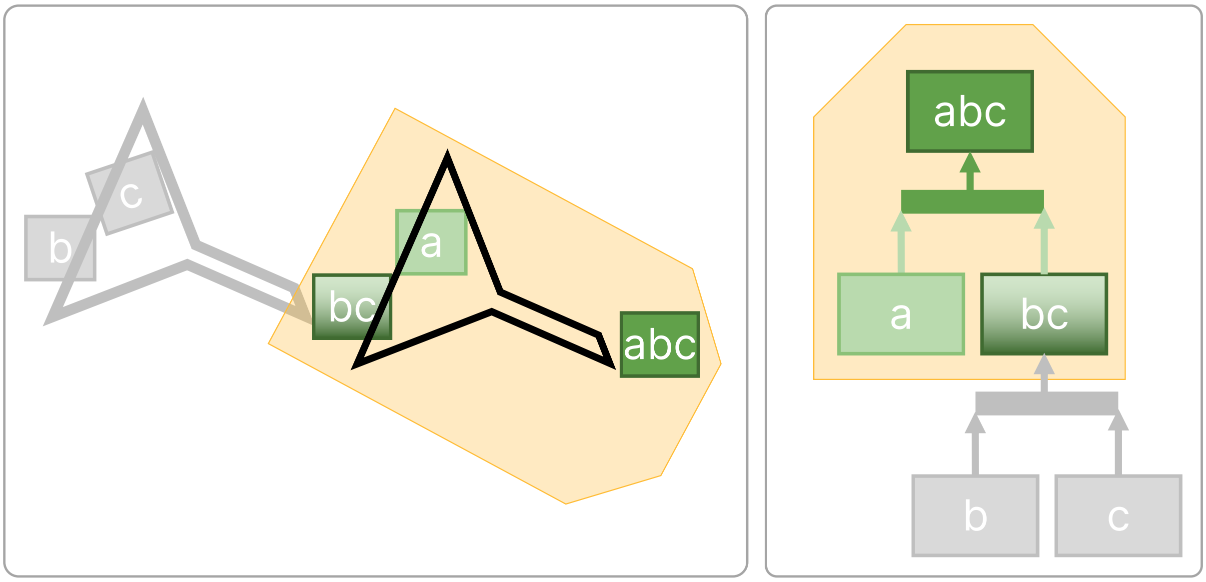


Figure 12: Example of the final stage in the construction history

The stricter (assembly) structure

Engineers building a BOM or WBS will want their breakdown to be both complete (to cover everything relevant) and for the components to be separate, disjoint. So, together these make components a kind of inventory (or simple single level bill of materials) of the composite. Where a good inventory must be complete: everything must show up somewhere. But it must also be judicious: nothing should show up more than once. Thus, the inventory should cover the composite short of overlap: every component should be disjoint from every other component – and the components should cover the composite completely.

The strict atomic compositions have exactly these properties. The components cover the whole composite, and the components are separate, disjoint. We start by looking at their structure more carefully. Breakdowns are often multi-level, so we then look at how these atomic compositions can be assembled into a hierarchy, which we have called a molecular composition, with the same properties.

The simplest atomic assembly structure

To recap, in this structure, disjoint components are (logically) assembled/composed into a single composite whole. One can reify the composing operation as a composing relation – an n-to-1-ary relation that partitions the composite into components (where the elements of the n-ary side of the relation are unique and unordered, in effect, a plurality). This composing relation is called an atomic composition. As with all relations, this can be viewed as the relata linkage between the composite and its components – or as the linked relata including the composite and its components as visualised in Figure 13 – however, these cash out to the same as the linkage includes the relata. The composite is the fusion of the components – furthermore, the composition (relation) can be dissected into corresponding part relations; one for each component-composite pair. This is also visualised in Figure 13.

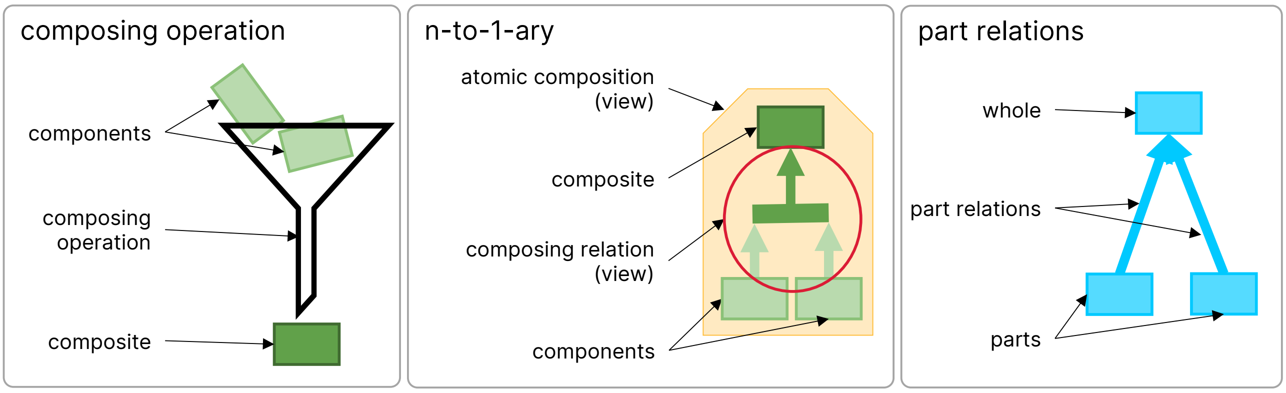


Figure 13: Three ways to visualise a simple atomic assembly structure

Formal constraints on the components being assembled

We have made a link between fusion and composition. In classical mereology a principle of unrestricted fusion is adopted, where any parts can be fused into a whole. But one does not have to adopt this, one can place restrictions on what can be fused (either in the underlying structure or in this structure).

Firstly, the association with fusion brings a constraint. Fusion comes with an in-built constraint, it ensures that the components *cover* the composite; in other words, there is no part of the composite that does not overlap one or more of the components. So the partial cover, partial separation and full separation pairs in Figure 14 cannot be composed into x, cannot have x as their composite. This is a logical consequence of the link to fusion (from the underlying part structure).

Secondly, we have a strictness constraint on the components. They need to be *pair-wise disjoint* – in other words, no pair can share a common part. Hence the overlapping and inside pairs in Figure 15 cannot be strictly atomically composed – though they could be non-strictly composed.

Tiling is a geometric procedure with the slogan 'no gaps, no overlaps' – which clearly applies here. This suggests another formal way of looking at this. As a kind of mereological (that is, sub-metrical) tiling constraint. Where, formally, a tiling is a collection of disjoint open regions, the closures of which cover the selected region, typically the whole plane (for more details see e.g. (Stein et al., 1994)).

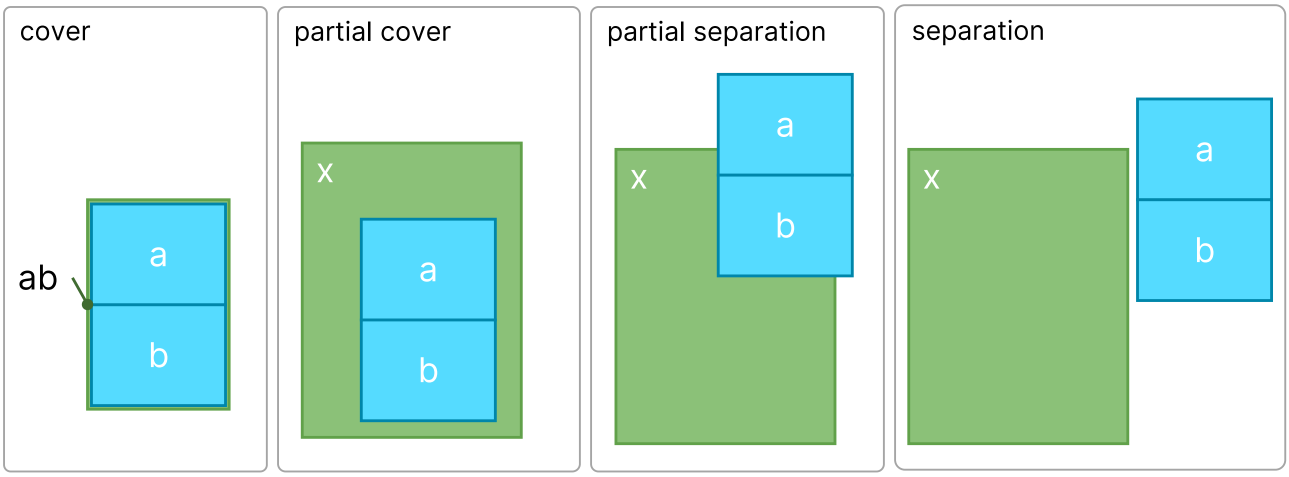


Figure 14: Cover and ways to not cover

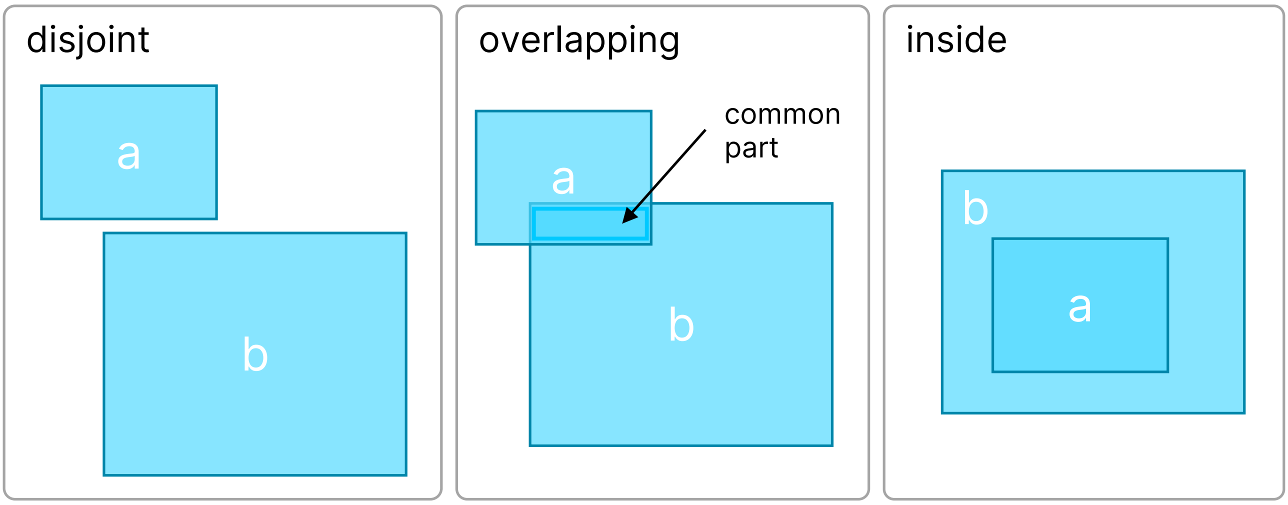


Figure 15: Disjointness and ways to not be disjoint

The requirement that the atomic components cover the atomic composite can be expressed as the requirement that the fusion of all the atomic components is the atomic composite whole. We do not formally express this requirement here.

The requirement that the components do not overlap (are disjoint) is the requirement that no two components' entities have a common part or equivalently that any two components of a composition are mereologically disjoint:

(P13) ∀x ∀y ∀z ((isAtomicComponentOf(x,y) ∧ isAtomicComponentOf(z,y) ∧ ¬ (x=y)) → disjointFrom(x,z))

Material constraints on the components being assembled

The constraints to fusion may be material as well as formal; one that only allows some types of object to be so composed. (Fine, 1999, p. 72) describes one kind of system that emerges from imposing a material constraint (his system of embodiment):

> "The majority of material objects [...] will submit to a hierarchical division into parts. Just as a car will have an engine, a chassis, and a body as immediate parts (these being the components of the rigid embodiment that is the current manifestation of the car), these immediate parts will themselves have further immediate parts, and so on all the way down until we reach the most basic forms of matter. Thus a material object will be like a set, with its hierarchical division into members, members of members, and so on.

In this Finean world, only some of the objects participate in the composing hierarchy – so only some objects can be composed into whole composites. If one pursues this route, then one needs to make the material constraints explicit – as Fine does.

Unique compositions and decompositions?

Once one has settled on the constraints, formal and material, one knows which pluralities of objects qualify for use as components in a composing relation. Given one of these qualifying pluralities then a natural assumption is that its members' fusion uniquely composes a composite whole. In inventory-speak, each inventory is for a single unique object.

One can then ask questions about, in terms of the TLO Survey (Partridge et al., 2020, sec. 4.2.1.1), whether the child-arity and parent-arity are single or multiple. Here we just consider the atomic compositions. Later, when we have introduced hierarchies, we ask the same questions relative to a hierarchy.

The first relation to consider is the (atomic) composing relation. Within the scope of a composition, this is defined as having single parent-arity and of multiple child-arity.

The next relation to consider is the participation (as composite or component) relation. In terms of child-arity and parent-arity, this is whether an object can participate as a component in multiple atomic compositions (parent-arity) and whether an object can participate as a composite in multiple atomic compositions (child-arity) – see Figure 16.

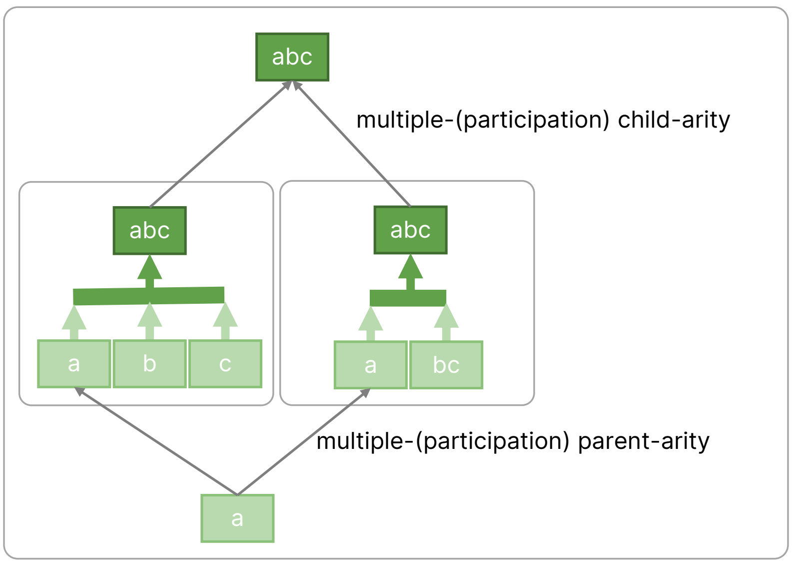


Figure 16: Participation (as composite or component) relation

Earlier (see Figure 10) we established that with unrestricted fusion an object can participate in multiple compositions. In the survey's terms, this would be a case of multiple child-arity. This would seem to be an essential feature of composition.

This raises the question whether (symmetrically) there could also be single, unique, decompositions, where each composite whole uniquely decomposes into a plurality of components. Whether, in the survey's terms, the parent-arity is single or multiple.

Fine, in the next paragraph after the one quoted in the previous section, writes: "… this division into parts will be largely unique". So he envisages a structure where the participation (as composite or component) relation is largely unique – where the child-arity and parent-arity is largely single.

However, this is an unusual position. In practice, as (Wimsatt et al., 2007, p. 182ff) notes scientists break down natural objects in many ways – in fact, he suggests this multiple decomposition is the mark of complex natural structures. Similarly, engineers typically provide multiple breakdowns of complex systems. In this case, the decomposition is not unique.

When one proposes a system of compositions, one should be clear about these aspects of its formal structure.

In the same spirit of plenitude (and simplicity) that infuses classical mereology and set theory – noted in the TLO Survey (Partridge et al., 2020, sec. 3.2.1), one could say that, given a part relation in a domain, then every possible partition based upon it is also a composition – in other words, that every disjoint plurality of objects in the domain can form a composite. This would allow for every possible multiple composition. It would only remain to highlight the more interesting ones.

One atomic composition atop another

Atomic compositions can be atop one another – where the composite of one composition is a component of another. We call this the atop relation. It is a relation between the compositions though it reflects a relation between their contents – as shown in Figure 17. This relation induces a partial order over the compositions.

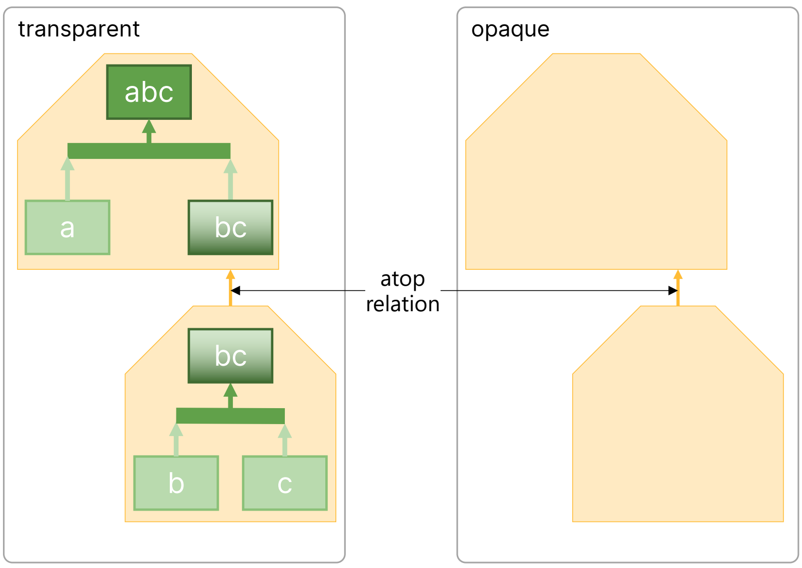


Figure 17: Atop relation

These atop relations can be seen as a composing sequencing relation, relating one composing operation to another to tie together the compositional history – as shown in Figure 18.

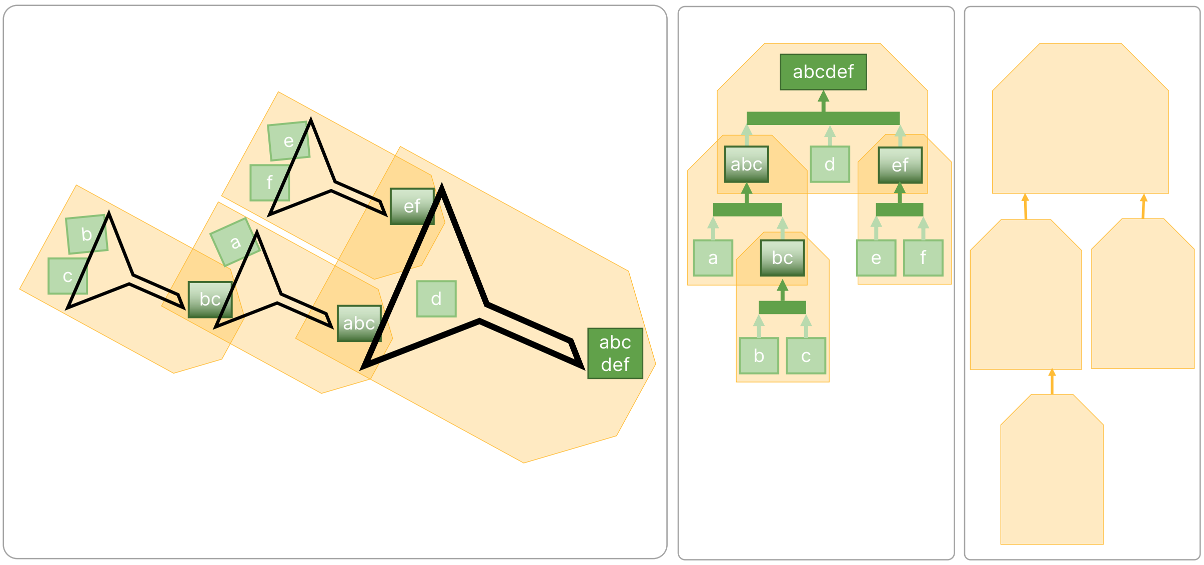


Figure 18: Example compositional history

This also shows a composition with multiple atop relation children – one corresponding to each of the possible compositions the element can appear as a composite – so it can have multiple child-arity. And, though not shown, a composition can have multiple atop relation parents – one corresponding to each of the possible compositions the element can appear as a component. In other words, the atop relation can have multiple parent-arity.

The atop relation can be seen (from a graph-theoretic perspective) as a branch re-composition. Where the branches are the histories, paths that trace out a tree. And the atop structure is the directed graph that is walked to produce these paths – hence a branch re-composition.

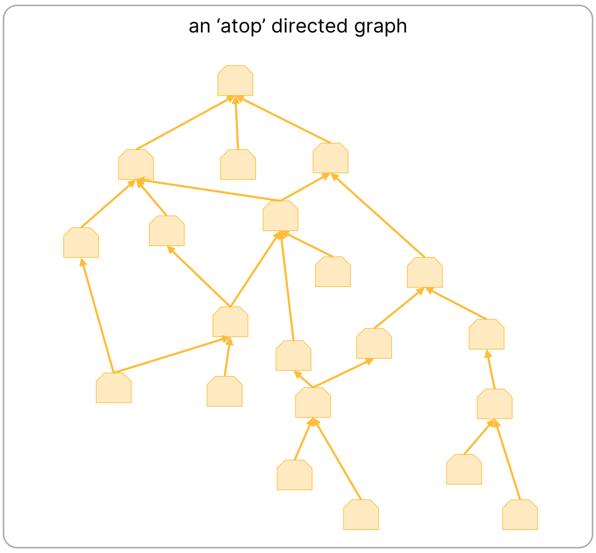


Figure 19: The branch recomposed atop directed graph

Atomic immediate parts

If one reads the atop relation as relating the composites of its related atomic compositions, then this is a kind of part relation – one that tracks only the underlying part relations between the composites. Read this way, the atop relation mereology has several distinctive features. For example, it is a mereology of '*immediate parts*'. Where an immediate part has no intervening mediate parts – as shown in Figure 20. This distinction of immediate and mediate parts goes back to (Husserl, 1970). The structure is (unlike normal part relations) antitransitive (if a is part of b and b is part of c, then a is not part of c). So in Figure 18, the lowest level atomic composition cannot be atop related to the top level composition.

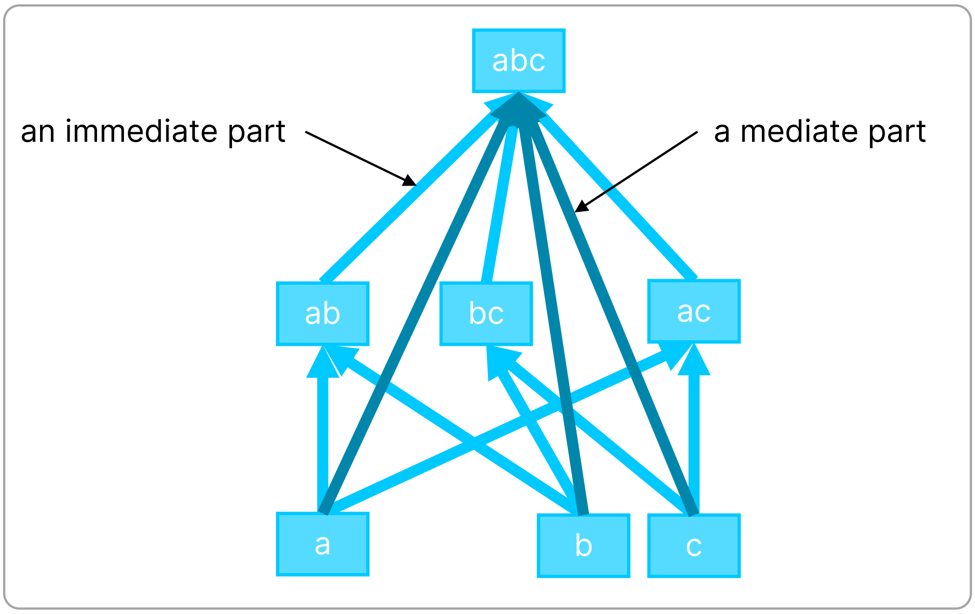


Figure 20: immediate and mediate parts

Dividing and combining atomic compositions

Atomic compositions can be divided and combined. The division involves taking a subset of the composition components and replacing it with another atomic composition – which gives two compositions one atop the other. Combining is effectively the reverse – as shown in Figure 21. One takes two compositions, one atop the other, and combines them, removing the overlapping element.

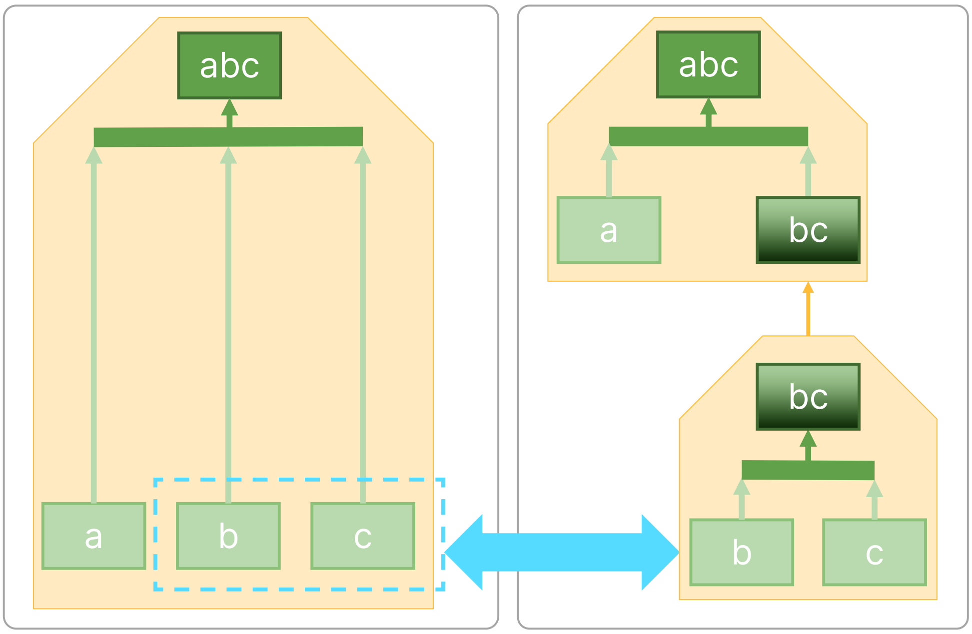


Figure 21: Dividing and combining atomic compositions

One can see these operations as shifting the path chosen through the construction history between the component leaves and the composite root – as shown in Figure 22.

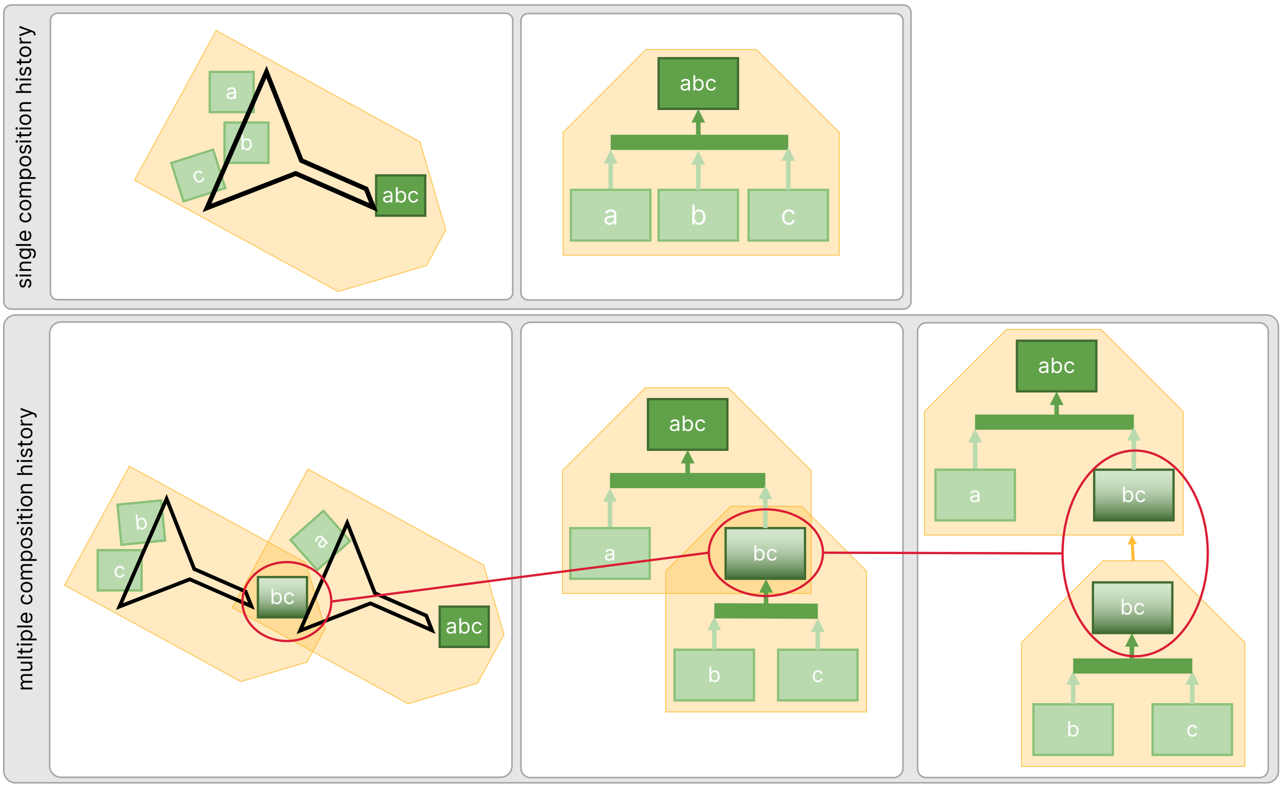


Figure 22: Example shifting the path chosen through the construction history

Another way to look at this is as navigating the inherent transitivity in the underlying sibling part hierarchy. Under transitivity if b and c are parts of bc and bc is part of abc, then b and c are parts of abc.

Hierarchies of strict atomic compositions

We now have an informal sketch of the structure and options for atomic compositions. With these in hand, we can build hierarchies of them, what we call *molecular compositions*.

Assembling atomic compositions into a hierarchy

The process of joining together some atomic compositions into a 'molecular' hierarchy composition is simple.

Firstly, we use the notion of 'atop' to characterise molecular compositions built from two atomic compositions where one is atop the other (in other words, where the atomic composite of one is also an atomic component of the other). Consider the molecular composition that results from joining these two. It has the two atomic compositions as atomic constituents. One can characterise this through the single atop relation that binds it together. More generally, subject to other constraints, associated with each atop relation, there is a doubleton molecular composition with two atomic constituents. If the atomic compositions being joined are disjoint, then the resulting doubleton molecular composition is too – through the shared element that 'supports' the atop relation.

atop(x,y) means that atomic composition x is atop atomic composition y.

atopAt(x,y,z) means that atomic composition x is atop atomic composition y and z is a component of x and the composite of y.

(P14) $\forall x\ \forall y\ (atop(x,y) \equiv (AtomicComposition(x) \wedge AtomicComposition(y) \wedge \exists z\ (isAtomicComponentOf(z,x) \wedge isAtomicCompositeOf(z,y))))$

MolecularComposition(x) means x is a molecular composition.

atomicJoinedInto(x,y,z) means that (the composite of) atomic composition x and (a component of) atomic composition y are joined into (doubleton) composition z.

isAtomicConstituentOf(x,y) means that x is an atomic constituent of y.

When two atomic compositions are in the atop relation, there is a molecular composition (though only the doubletons):

(P15) $\forall x\ \forall y\ (atop(x,y) \rightarrow \exists z\ (MolecularComposition(z)))$

The above is weak and we use joinedInto to express the stronger joining of the atomic compositions in a molecular one.

(P16) $\forall x\ \forall y\ (atop(x,y) \rightarrow \exists z\ (MolecularComposition(z) \wedge atomicJoinedInto(x,y,z)))$

That the joined atomic compositions are atomic constituents of the molecular composition resulting from the joining operation can be expressed as:

(P17) $\forall x\ \forall y\ \forall z\ (atomicJoinedInto(x,y,z) \rightarrow (isAtomicConstituentOf(x,z) \wedge isAtomicConstituentOf(y,z)))$

In fact, we can define isAtomicConstituent in terms of joinedInto:

(P18) $\forall x\ \forall y\ (isAtomicConstituentOf(x,y) \equiv \exists z\ (atomicJoinedInto(x,z,y) \vee atomicJoinedInto(z,x,y)))$

Then we need to extend some of the atomic vocabulary to cover molecular compositions. The elements of this molecular hierarchy can be classified in three ways, based upon their ‘roles’ in the constituent atomic compositions. A single top ‘composite’ element – which belongs to only one atomic constituent and is a composite in that atomic composition. This constituent we call a top atomic constituent. Joined elements, which each belong to two atomic constituents, where it is a composite in one of these atomic compositions and a component in the other. All component elements that are not joined are free elements. This classification enables us to say that one can join a molecular composition to an atomic composition at its free elements to build a new molecular composition.

Atomic constituents that have one or more free component elements are called open atomic constituents. Atomic constituents that are not open are called closed atomic constituents. Atomic compositions are their own atomic constituents. These new classifications are visualised in Figure 23.

isCompositeOf(x,y) means that x is the composite element of (the composition) y.

isFreeComponentOf(x,y) means that x is a free component of y.

isJoinedElementOf(x,y) means that x is a joined element of y.

isTopAtomicConstituentOf(x,y) means that atomic composition x is a top atomic constituent of y.

isOpenAtomicConstituentOf(x,y) means that atomic composition x is an open atomic constituent of y.

isClosedAtomicConstituentOf(x,y) means that atomic composition x is a closed atomic constituent of y.

Every molecular composition has a single top atomic constituent:

(P19) ∀x ∀y ∀z ((MolecularComposition(x) ∧ isTopAtomicConstituentOf(y,x) ∧ isTopAtomicConstituentOf(z,x)) → y = z)

It follows that every molecular composition has a unique composite:

(P20) ∀x ∀y ∀z ((MolecularComposition(x) ∧ isCompositeOf(y,x) ∧ isCompositeOf(z,x)) → y = z)

The composite of a molecular composition is the atomic composite of its top atomic constituent:

(P21) ∀x ∀y ∀z ((isCompositeOf(x,y) ∧ isAtomicConstituentOf(x,z)) → isTopAtomicConstituentOf(z,y))

It follows that the composite belongs to a single constituent:

(P22) ∀x ∀y ∀z ((isCompositeOf(x,y) ∧ isAtomicConstituentOf(y,z) ∧ isAtomicConstituentOf(u,z)) → u=y)

We can break down the definition of free component above into 3 propositions possibly overlapping. Firstly, free components in a molecular composition are atomic components in one of its atomic constituents:

(P23) ∀x ∀y (isFreeComponentOf(x,y) → ∃z (isAtomicConstituentOf(z,y) ∧ isAtomicComponentOf(x,z)))

Furthermore, a free component belongs to only one atomic constituent:

(P24) ∀x ∀y ∀u ∀v ((isFreeComponentOf(x,y) ∧ isAtomicConstituentOf(u,y) ∧ isAtomicConstituentOf(v,y) ∧ isAtomicComponentOf(x,u) ∧ isAtomicComponentOf(x,v)) → u = v)

The atomic constituent to which a free component belongs is in fact an open atomic constituent:

(P25) ∀x ∀y ∀z ((isFreeComponentOf(x,y) ∧ isAtomicConstituentOf(z,y) ∧ isAtomicComponentOf(x,z)) → isOpenAtomicConstituentOf(z,y))

We can define joined elements as above:

(P26) ∀x ∀y (isJoinedElementOf(x,y) ≡ ∃u ∃v (isAtomicConstituentOf(u,y) ∧ isAtomicConstituentOf(v,y) ∧ ¬ (u = v) ∧ isAtomicCompositeOf(x,u) ∧ isAtomicComponentOf(x,v)))

Open atomic constituents are atomic constituents with free components:

(P27) ∀x ∀y (isOpenAtomicConstituentOf(x,y) ≡ (isAtomicConstituentOf(x,y) ∧ ∃z isFreeComponentOf(z,x)))

Closed atomic constituents can also be defined as above:

(P28) ∀x ∀y (isClosedAtomicConstituentOf(x,y) ≡ (isAtomicConstituentOf(x,y) ∧ ¬ isOpenAtomicConstituentOf(x,y)))

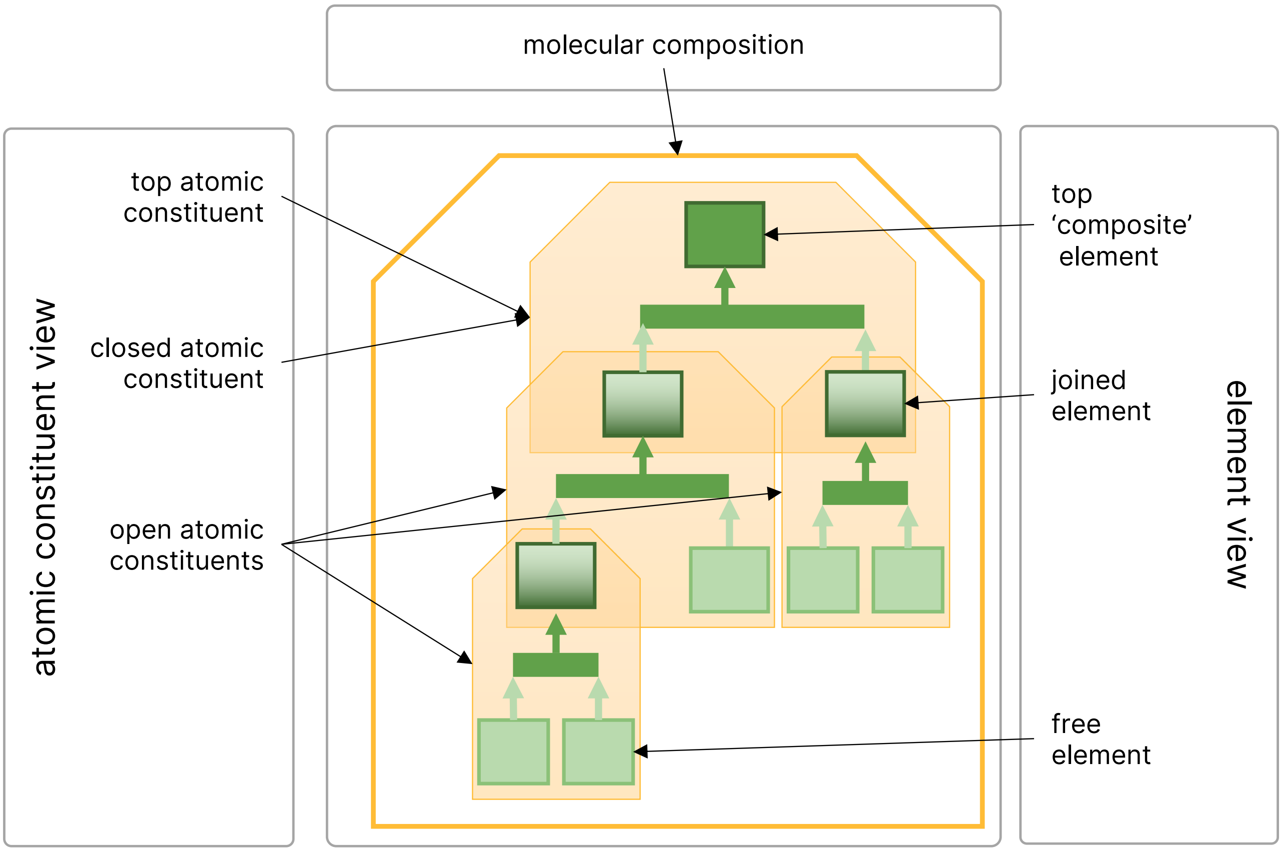


Figure 23: Molecular composition views and classifications

With these definitions, we can define a general joining relation using atop. Given any composition, we can join another atomic composition to open atomic constituents if they are in the right kind of atop relation. The joined atomic composition becomes a constituent of the new molecular composition. In this way, step by step, atomic constituent by atomic constituent, we build up all possible molecular hierarchies. A simple example is shown in Figure 24, where atomic compositions are added one by one (we use a '+' sign in the name to mark this addition). This process makes clear that compositions can be characterised by their atomic constituents (where atomic compositions are deemed to have just themselves as atomic constituents).

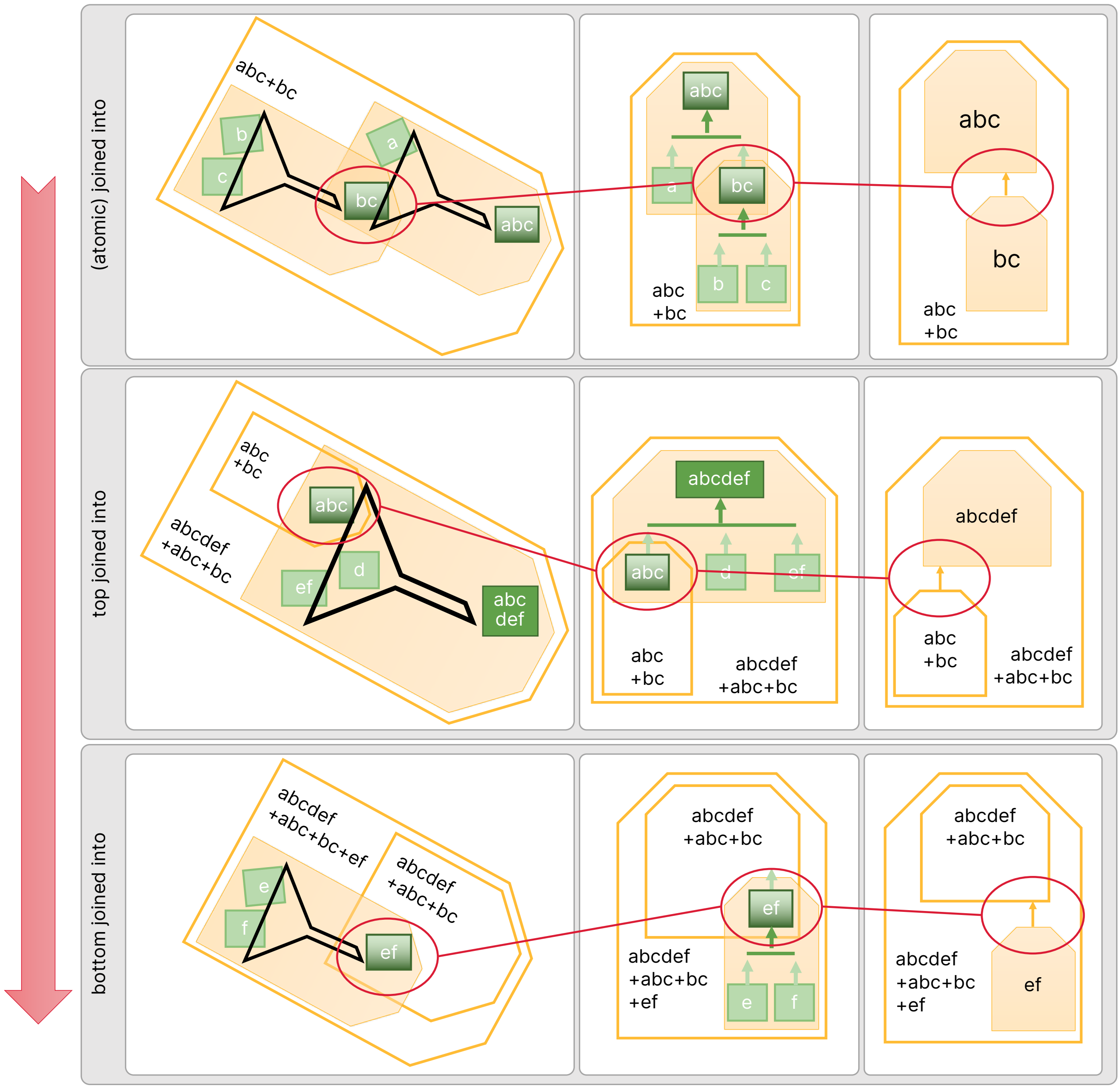


Figure 24: Step-by-step construction of molecular compositions

Composition(x) means x is a composition.

topJoinedInto(x,z,w) means that the composition x is joined at the top to atomic composition z to build composition w.

Compositions can be joined with atomic compositions at the top into new compositions:

(P29) ∀x ∀y ∀z ((Composition(x) ∧ isTopAtomicConstituentOf(y,x) ∧ AtomicComposition(z) ∧ atop(z,y)) → ∃w (topJoinedInto(x,z,w) ∧ ∀u (isAtomicConstituentOf(u,x) → isAtomicConstituentOf (u,w)) ∧ isAtomicConstituentOf(z,w)))

In these cases, the 'new' top constituent, the top constituent of the result of joining is the atomic composition used in the joining.

(P30) ∀x ∀y ∀z ∀w ((Composition(x) ∧ isTopAtomicConstituentOf(y,x) ∧ AtomicComposition(z) ∧ atop(z,y) ∧ topJoinedInto(x,z,w)) → isTopAtomicConstituentOf(z,w))

bottomJoinedInto(z,x,w) means that atomic composition z is joined to the bottom of composition x to build composition w.

Compositions can be joined with atomic compositions at the bottom into new compositions:

(P31) ∀x ∀y ∀z ((Composition(x) ∧ isOpenAtomicConstituentOf(y,x) ∧ AtomicComposition(z) ∧ atop(y,z)) → ∃w (bottomJoinedInto(z,x,w) ∧ ∀u (isAtomicConstituentOf(u,x) → isAtomicConstituentOf (u,w)) ∧ isAtomicConstituentOf(z,w)))

In these cases, the 'new' top constituent is the 'old' top, that is the top atomic constitutent of the composition used in the joining.

(P32) ∀x ∀y ∀z ∀w ∀u ((Composition(x) ∧ isOpenAtomicConstituentOf(y,x) ∧ AtomicComposition(z) ∧ atop(y,z) ∧ bottomJoinedInto(z,x,w) ∧ isTopAtomicConstituentOf(u,x)) → isTopAtomicConstituentOf(u,w))

Compositions resulting from joining are made of the constituents of the compositions joined:

(P33) ∀x ∀y ∀z ∀u (joinedInto(x,y,z) → (isAtomicConstituentOf(u,z) ≡ (isAtomicConstituentOf (u,x)) ∧ isAtomicConstituentOf(y,z))

Compositions are atomic compositions or molecular compositions resulting from the general joining operation:

(P34) Composition(x) ≡ (AtomicComposition(x) ∨ MolecularComposition(x))

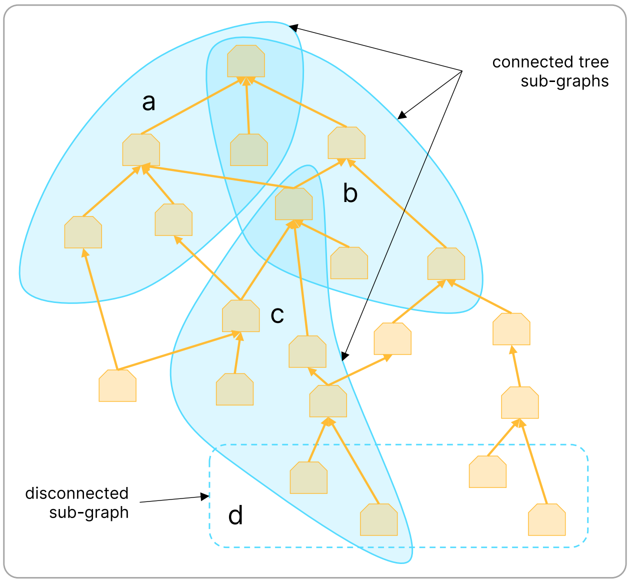


Figure 25: visualising connected tree sub-graph histories

Another way to visualise this is as taking a connected tree sub-graph of the atop relation graph, where every branch relates to different components – see Figure 25. As this last way of visualising makes particularly clear, molecular compositions can be characterised (in part) by the atop relations they involve.

Criterion of identity

The earlier comments translate into a criterion of identity for compositions – that of being constituted by the same atomic constituents. Two molecular compositions with the same atomic constituents are the same composition. Two atomic compositions with the same atomic constituents (that is, themselves) are plainly the same composition.

> Based on the formalism introduced already, the criterion for identity of compositions can now be formulated as follows:
>
> (P35) ∀x ∀y ∀z ((Composition(x) ∧ Composition(y) ∧ ∀z (isAtomicConstituentOf(z,x) ≡ isAtomicConstituentOf (z,y))) → x=y) (Composition Criterion of Identity)

A new 'mereology'

For those familiar with the history of mereology, the connected graph visualisation suggests the possibility of a mereology of connected regions. One similar to Whitehead's region mereology in variously (Whitehead, 1916), (Whitehead, 1919), (Whitehead, 1920), (Whitehead, 1929) and the associated (De Laguna, 1922) and (Clarke, 1981). This should not be surprising as, for example, (Cotnoir et al., forthcoming) notes how Whiteheadian mereologies can arise through relativisation. This is interesting as Whitehead was aiming for a structure of well-behaved regions, and this suggests that compositions may behave in a similar fashion.

Whitehead originally used 'extends over' as his primitive but revised this after Laguna's paper to use 'connection' and Clarke followed suit. The atop relation could be regarded as a similar kind of connection relation. There is a difference, but it is not relevant – Whitehead's theory was atomless and this is atomic – being built from atomic compositions. Another approach would be to define a part relation between compositions based upon all the atomic constituents of the part also being parts of the whole.

This structure allows for mediate parts (unlike the earlier atomic atop structure). Its fusion operation – joins – is not unrestricted. It only joins atop-related compositions. This ensures that the result is atop-connected (in a similar fashion to Whitehead ensuring all his regions are connected). It also has simple boundary – connection conditions. Under the hood, the atop-connected compositions share an object.

Extending strictness to molecular compositions

The extension from atomic to molecular is simple. Strict molecular compositions are those whose atomic constituent compositions are all strict. One can generalise the definition. A composition is strict if all its atomic constituents are strict.

Associated parts – extending dissection to molecular compositions

One can dissect the associated parts of a molecular composition by collecting the dissected parts of all its atomic constituents. Molecular compositions share many of the properties of atomic compositions, making these general composition properties.

For example, dissected part relations do not appear twice in the same molecular composition, though they can appear many times in different compositions, molecular or atomic. The dissected part relations uniquely define the molecular composition, there is no other composition with exactly the same relations – in this sense, the relations are a signature for the molecular composition – and more generally for compositions.

The mode of construction of molecular compositions preserves the 'mereological harmony' of their atomic constituents. So there is no dissected relation that is the 'wrong-way around'.

Compositional immediate parts

We mentioned earlier that the atop relation induced an immediate part structure over the atomic compositions. The molecular compositions (due to their mode of construction) have a parallel immediate part structure. One way of visualising this is starting with all the part relations (which, if dense, might not have any immediate parts) and pick out sub-sets of these that correspond to molecular compositions. These will only contain connected immediate parts.

Material constraints on the component hierarchy

The material constraints on the atomic compositions are also inherited by the hierarchies – as only the compositions allowed by those material constraints can participate in the hierarchy.

Additional material constraints can be added if required. Though, in practice these seem to be local to specific hierarchies rather than global. Where, for example, a location breakdown structure may restrict its nodes to locations, whereas a systems breakdown structure might restrict its nodes to systems.

Overlap for molecular component hierarchies

There are various ways two molecular compositions can overlap. The most liberal is where two compositions share elements: this does not imply they share any atomic constituents. Atop is a specialised case of this. A less liberal, coarser grained, way is where two compositions share atomic constituents: this does not imply that the full overlap is a composition. An even less liberal, coarser grained, way is where two compositions' overlap is another composition. These are illustrated in Figure 26.

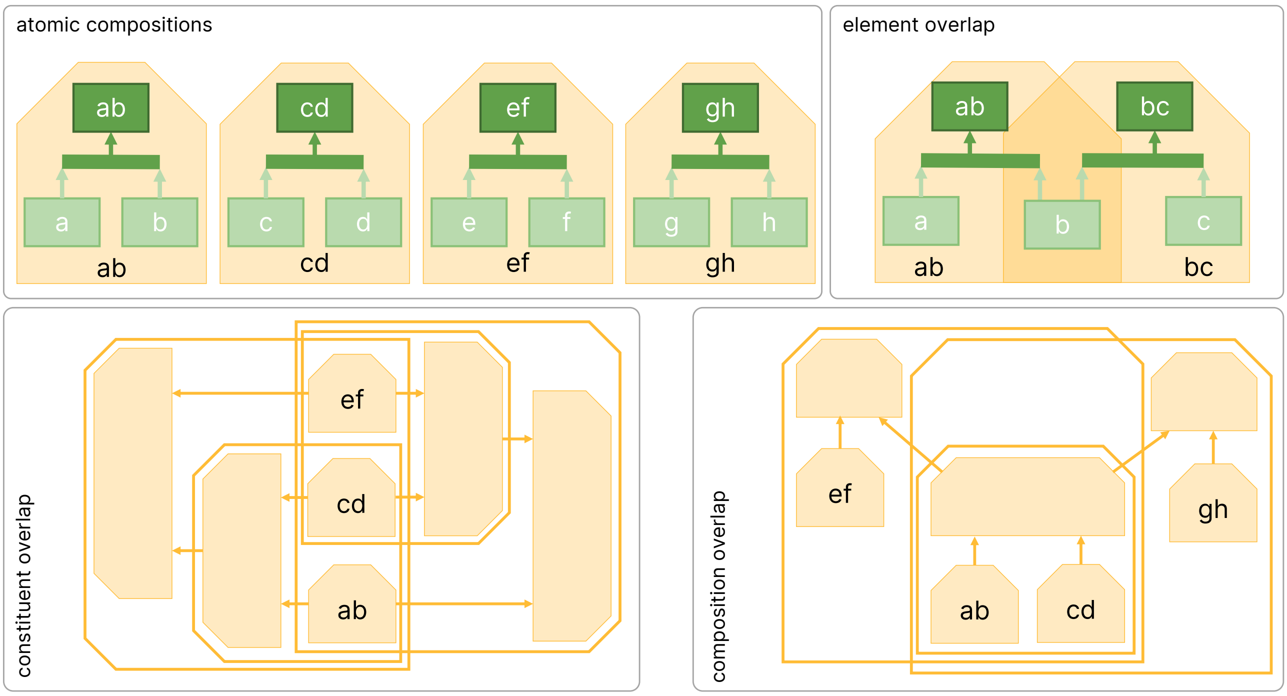


Figure 26: Overlapping molecular component hierarchies

Refining and collapsing constituents in a component hierarchy

The dividing and collapsing of atomic compositions form the basis for the refining and collapsing of molecular hierarchies. When an atomic constituent of a molecular hierarchy (an atomic composition) is divided this results in a refined molecular hierarchy. When the reverse happens and two atomic constituents are combined, the result is a collapsed hierarchy. In the first case, there is an insertion of an element (refinement) and the second a removal of an element (collapse) – see this visualised in Figure 27. One can concatenate a series of refinements or collapses – these then zoom us in and out of the underlying part structure. Interestingly, it is feasible to zoom in and then zoom back out to a different hierarchy.

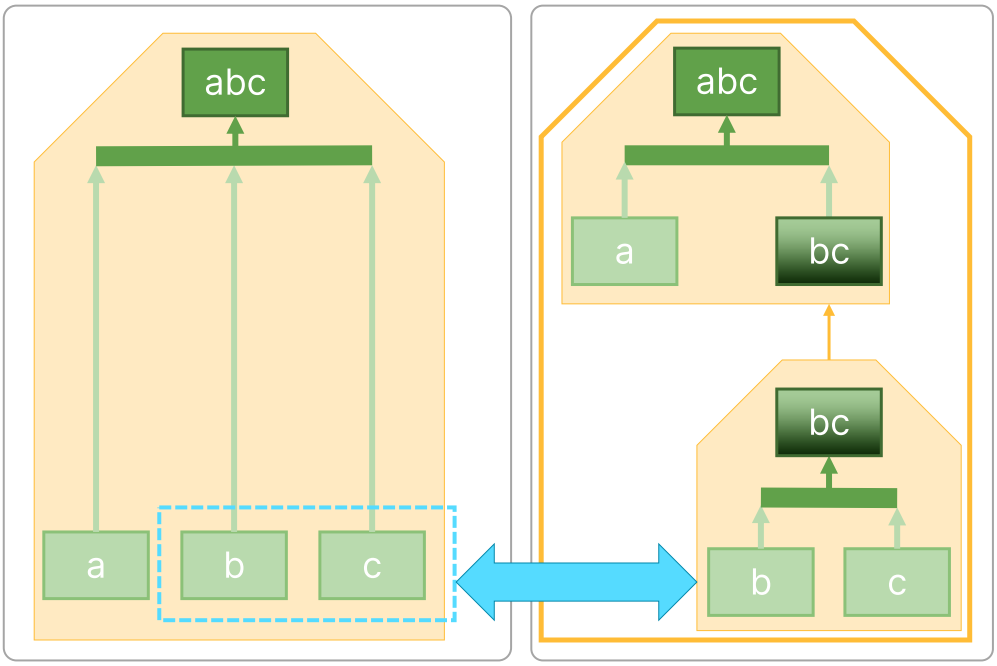


Figure 27: Refining and collapsing constituents example

More specifically, the process of reduction involves collapsing (a process described above) a joined atomic constituent and into two joined atomic constituent in the composition hierarchy. The process of extension involves dividing (also described above) an atomic constituent. Obviously, there can be a series of these processes taking one from one hierarchy to another.

Generating composition structures

One can approach the question of formal generation (see the TLO Survey, Section 4.2.1.5 (Partridge et al., 2020) in a number of ways.

The way we favour would be, for each domain, to bind the part and composition relations, including their formal generations. Where every disjoint plurality of parts is input for an atomic composition and every atomic composition breaks down into part relations. If one adopts a plenitudinous approach to part relation generation, then this is inherited by composition. Given this one then (plenitudinously) formally generates every possible composition hierarchy.

There are other possible strategies. One could avoid formal generation and only recognise the compositions that are 'valid'. One could generate the part relations from the valid compositions. Whichever decision is made, it is good to make this explicit.

Relaxing the strict (de-)composing structures

In many cases, the strict structures make sense. In the case of physical assembly, the process of physical assembly involves putting disjoint parts together to make a complete whole. Similarly with physical tiling or tessellation, the individual tiles are disjoint and are combined to cover the whole floor. In fire safety engineering, the physical asset is similarly divided into fire zones.

However, there are cases where the strictness may seem too constraining. We look at two approaches, firstly at one of relaxing then at one of adaption.

Relaxing disjointness

There are cases where disjointness seems too strict. For example, consider a car whose systems breakdown includes a fuel system and an electrical system. The systems are largely disjoint. But at the lower levels they overlap. For example, the fuel pump is part of both the fuel system and the electrical system – as shown in Figure 28. There is some leeway for increasing the disjointness; one could argue that the pump is not really part of both the fuel system and the electrical system, and that it is only the pump head that is part of the fuel system and the electric motor that is part of the electrical system. But this only works so far, eventually there is some irreducible overlap; the shaft that links the motor to the pump head must be a part of both or nothing happens, and it is the load on the fuel pump that determines the draw on the battery.

One cannot argue that compositions are not necessarily disjoint. In the simple examples at the beginning of the report, we chose to filter out the non-disjoint compositions. So we have options to relax the disjointness constraint. A simple relaxation that still places a degree of constraint is one that, within an atomic composition, while the components can overlap, no components are parts of one another. (Schaffer, 2010) has a similar suggestion for his far grander tiling.

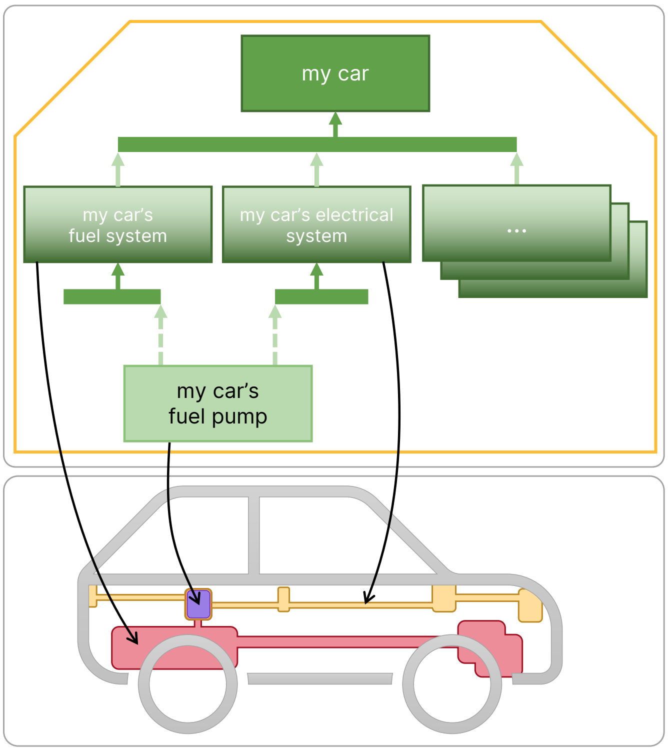


Figure 28: A simple disjoint relaxation example

The systems breakdown above follows this pattern. At each decomposition none of the components are part of each other (the limit case, where the components are improper parts of each other – in other words, the same – is formally excluded). And it illustrates the potential drawback. If the components overlap then as one drills down, it becomes more likely that the same component will appear twice. It also illustrates how one can limit the relaxation. It may only be the initial level of the breakdown – between fuel and electrical systems that relaxes the constraint. Further decompositions within these two systems can all be disjoint.

PairwiseNoPartAtomicComposition(x) means that atomic composition x satisfies the relaxed ‘no parthood’ constraint, i.e., no two components of x are in the whole-part relation.

(P36) ∀x (PairwiseNoPartAtomicComposition(x) ≡ (AtomicComposition(x) ∧ ∀y ∀z ((isAtomicComponentOf(y,x) ∧ isAtomicCompositeOf(z,x) ∧ ¬ (y=z)) → ¬ wholePart(z,y)))) (No Parthood for Atomic Compositions)

Under the standard construction of molecular composition hierarchies, ‘no parthood’ is not inherited. The overlapping part may turn out to be a component of multiple compositions at a lower level. No parthood can be extended to molecular component hierarchies.

PairwiseNoPartComposition(x) means that composition x respects ‘no parthood’, i.e., all its atomic constituents do.

(P37) ∀x (PairwiseNoPartComposition(x) ≡ (Composition(x) ∧ ∀y (isAtomicComponentOf(y,x) → PairwiseNoPartAtomicComposition(y)))) (No Parthood for Compositions)

There may be cases where the weaker ‘no atomic parthood’ seems more appropriate.

Accommodating partial cover

As our earlier discussions of composition have made clear, covering is an essential feature of the formal notion of composition. It seems odd to claim that only some of the components somehow compose the whole. Hence the importance placed on BOMs and WBSs being complete. However, when the composition is a decomposition analysis, the evidence is less clear. There seem to be cases where one only focuses on a part of the whole.

However, let us dismiss something similar. There are cases where one has a focus, but one acknowledges there is a remainder. Typically, one only classifies it as other. While the focus components in the composition may have been chosen for some reason, the remainder is what is left after these components have been exhausted. This is not really a failure of cover, it is more a decision not to analyse.

There are cases, where one has a focus on merely a part of the whole and make no acknowledgement there is a remainder. However, one would acknowledge that the breakdown is partial. It seems to us that the previous case is a good guide to what is happening. Once one acknowledges that the breakdown is partial – one is acknowledging that there is a remainder. In this case, one is just not bothering to show this in the reporting. Or maybe not showing directly.

Accommodating families

The fuel and electrical system example above was a breakdown of aspects of the whole where the aspect related to a part of the whole – so the breakdown of the whole into its aspects is a breakdown into parts, albeit overlapping. However, we often encounter a breakdown of the whole into multiple aspects, where each aspect covers the whole – where the aspectual nature is the structure of the breakdown. In these cases, there is a family of breakdowns, all sharing a common top composite.

As noted earlier by (Wimsatt et al., 2007, p. 182ff), multiple decomposition is the mark of complex structures. He also notes that less complex structures tend to have decompositions that "give spatially coincident boundaries for parts" whereas more complex structures "give spatially non-coincident boundaries for parts". This suggests that within these families of decomposition there is a range of shared elements. This being more likely for the less complex and less likely for the more complex – as in the more complex cases, the boundaries are more likely to be non-coincident.

Assessment framework

The preceding analysis provides a basis for assessing the general foundation of a composition data structure (or ontology). Here we build such an assessment framework. It can be regarded as an extension of the assessment in the TLO Survey (Partridge et al., 2020). The aim of this framework is to provide an overview of the situation rather than an exhaustive detailed review – though it can easily form the basis for this.

The first step in the assessment process is to determine whether there are any composition structures to assess.

For simplicity, we focus on the two clear cases identified at the start of the report:

- Is there an individual wholes-parts composition structure (mereonomy)?
- Is there a class super-subclass composition structure (taxonomy)?

The analysis of the structure is general, and so can be applied to both these cases, where they exist.

We ask about the timelessness of composition.

- Are the composition relations timeless (hence ontologically immutable)?
- Where the domain of the composition is not timeless (as with mereonomy), are there structures to handle replacement parts?

We ask about the sibling part structure.

- Does composition have a sibling part structure? If there is a sibling part structure:
  - Is this structure formalised? If so, how?
  - is mereological harmony between it and the composition structure enforced? If so, how is it enforced?

We ask about the basic formal structure of the relation:

- What is the minimum number of components allowed for a composition?
  - Are zero components allowed (introducing a null component object)?
  - Is a single component allowed (allowing for improper compositions)?

- Do the composition relations exclude repetition and order from the components?
- Do compositions have an immediate parthood structure?

We ask about the criterion of identity.

- Is there a criterion of identity for compositions?
- How does this relate to the criterion of identity for the sibling part relations?

We ask about the child- and parent-arity of the relations between elements and compositions.

- Are there multiple composition relations?
- Can an element be a component in multiple compositions (multiple child-arity)?
- Can an element be a composite in multiple compositions (multiple parent-arity)?

We ask about the formal constraints available for compositions.

- Can compositions be constrained to be disjoint? And is this optional?
- Can compositions be constrained to non-parthood? And is this optional?
- Can compositions be relaxed to allow partial cover? And is this optional?
- Can different constraints be applied to different tranches of the composition?

We ask about the material constraints available for compositions.

- Are there any global material constraints upon compositions?
- Are there any local material constraints upon specific compositions?

We ask about whether and how one can zoom in and out of the part structure using compositions.

- Can one divide and combine atomic compositions?
- Can one refine and collapse molecular compositions?

Finally, our approach to characterising combinations has been based upon the notion of a process of composing, starting with atomic compositions and then to the stepwise joining of these into molecular compositions. And then from this the processes of refinement and collapse. Along with this comes a series of ways to visualise the underlying structure, including composition histories. This has explanatory and structural advantages, as it leads naturally to the kind of composition structure we are seeking to characterise. Of course, we are making no claims that this is the only way to characterise this structure. But, it helps when proposing a composition structure to be able to explain how it emerges. Also, this particular way of explaining the structure can be used as a benchmark for assessing other explanations.

Further work

The focus in this paper has been on the component as an integral (dependent) part of the composite whole. This prepares the ground for further foundation work on the sibling notion of ‘replaceable part’ (see Appendix A). This will look at the classification of the parts; so where one wants to say a car of this type will have integral components including an engine and four wheels. Where these integral components will have replaceable parts that are initially of some type.

We also think it may be useful to explore the possibility of nesting compositions inside compositions, something we did not have time to do here.

Summary

The paper has given an in-depth examination of the rich formal structure, providing a reasonable guide to what is required to produce a reasonably accurate representation – in particular, one that is sufficiently accurate to support interoperability.

It has also provided an assessment framework, a supplement for the Ontology Assessment Framework in (Partridge et al., 2020), which gives a structured way to both assess existing models as well as design new models.

Appendices

Appendix A – Integral and replaceable parts

As we note in the body of the report there is a sense of a part as an integral component, in which the part belongs to the whole. We also note there is an alternate sense of a replaceable (or temporary) part that can not only be replaced but also be used in multiple wholes.

This distinction seems to rely to some extent on the notion of time; making a distinction between timeless and temporary parts. As such it seems to apply more to domains that are embedded in time, such as individuals and their whole-part relations. For domains that are, in a sense, timeless (such as sets) then every part is, in a sense, integral. For more details on the kind of structures that arise in this case see (Partridge et al., "Formalization of the classification pattern: survey of classification modeling in information systems engineering", 2016).

We highlight these two senses by noting their appearance in the literature. (Partridge, Business objects: re-engineering for re-use, 1996), (Partridge, What is Pump Facility PF101? A Study in Ontology, 2002) and (West, "Replaceable parts: A four dimensional analysis", 2003) provide an extensional 4-dimensional analysis of the distinction. (Fine, "Things and their parts", 1999) provides a radical Aristotelian analysis. The early sections (Simons, Parts: A Study in Ontology, 1987), (West, The Data Management Guide: Developing High Quality Data Models: Volume 3, 1994) and (Simons et al., "Aspects of the Mereology of Artifacts", 1996) provide useful introductory background.

Parts: A study in ontology

(Simons, Parts: A Study in Ontology, 1987), the standard textbook on the ontology of parts, raises the problem of parts in flux:

> To a first approximation only, and avoiding for the moment questions of the identity of things in flux (i.e. things which lose and gain parts), we might say that, at any one time, manufacturers components form the base for machines, while cells form a base for organisms. One machine can be part of another (as a carburettor is part of an automobile),

but different machines differ by at least a component, as different organisms differ by at least a cell. (pp. 44-45)

A wall may survive partial destruction of the stones making it up, but not their total destruction, whereas a wall may be totally destroyed and its stones remain unscathed. Of course, an object which lives through flux in parts may continue to exist although all its previous components are destroyed, so long as its present components remain reasonably intact. But a constituted object may in general be destroyed with little or no destruction of its components, so the asymmetry of constitution is guaranteed. The point demonstrates another aspect of the practical and theoretical importance of objects in flux: theoretically, they present a special case; practically, an object in flux has a further kind of survival potential. (p. 240)

When we turn to artefacts, it is again original parts which are the best candidates for being essential. It is hard to conceive how an object could have as essential a part which was attached at some time after the object had come into being, since the part could quite easily have ceased to exist in the meantime, without prejudice to the already existing artefact. In the case of an object assembled from pre-existing components, one may claim that, at least for some of the more important parts, that object would not have existed had this not been part of it. (p. 271).

Developing high quality data models – 1994

(West, The Data Management Guide: Developing High Quality Data Models: Volume 3, 1994) makes the distinction between timeless parts and changeable parts – the parts in flux from above – in exactly the sense we are after:

Facility is a term we use to describe the service performed by particular pieces of *equipment*. Whereas you find *equipment* numbers stamped on pieces of *equipment*, *facility* identifiers (often referred to as tag numbers) are found on drawings and in the control room, and are the way that operators normally refer to their plant. The *installation* template is shown in Figure 2.9 below.

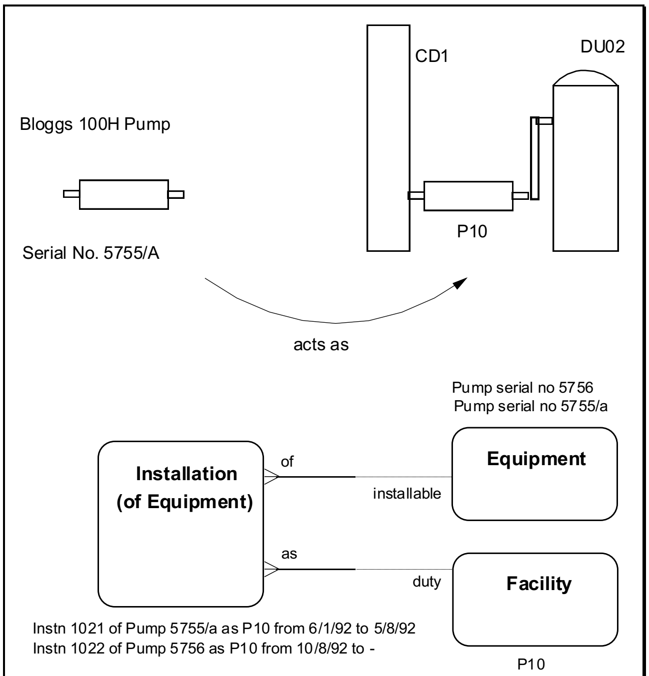


Figure 2.9: Installation template.

In the model above the *facility*, P10, is shown as the duty served by two *installations*. The first shows pump 5755/a has been installed to act as P10 from 6/1/92 to 5/8/92. The second shows pump 5756 has been installed from 10/8/92 and has not been removed. The pump 5755/a may, after being repaired, be installed to act as another *facility*, or be reinstalled to act as P10.

Aspects of the mereology of artifacts

Peter Simons, with Charles Dement returns to the topic in (Simons et al., "Aspects of the Mereology of Artifacts", 1996). They provide a good picture of how assemblies work, but do not quite capture exactly the sense we are after.

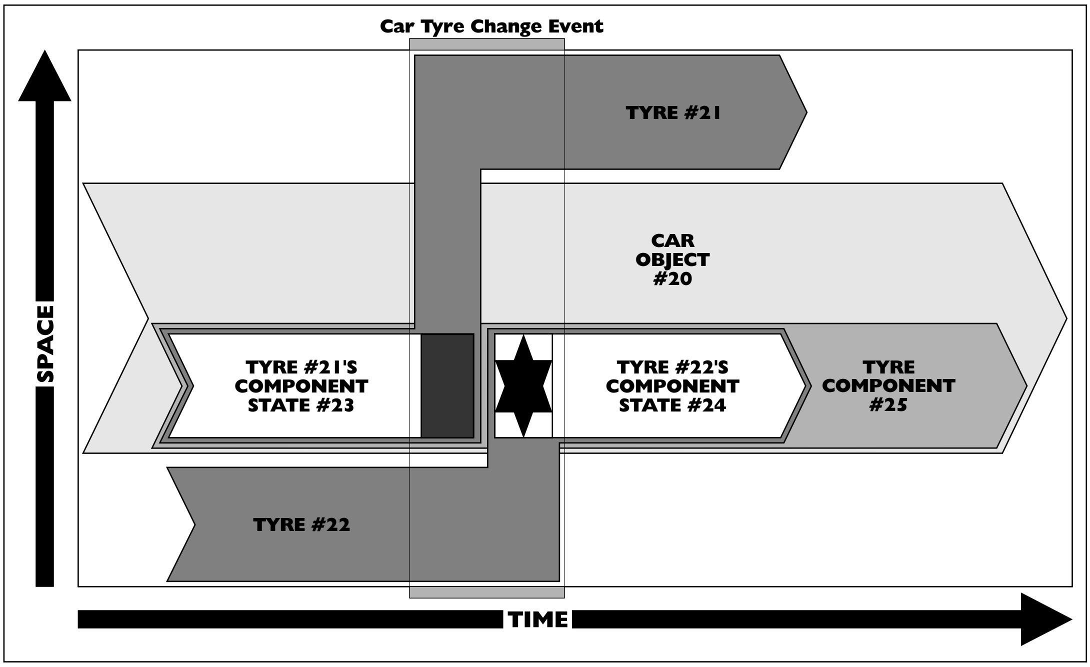
> Firstly, something can be a component because it is manipulated as a unit during the assembly or manufacture of the artifact. There are two kinds of what we shall call assembly components. The first are parts which are not themselves assembled but which come to the assembly process ready to be put together with others. We call these simple assembly components or assembly atoms. … Next, there are assemblies, which are assembly components made up out of two or more assembly atoms. The carburettor of a car is an assembly, for example: it has a float bowl, float, jet, butterfly valve, springs, washers and so on. The float itself is a subassembly, that is, an assembly which is an assembly component of an assembly, here consisting of the float body, pins, hinges etc. The carburettor is itself a subassembly, being an assembly component of the engine. When assemblies are mated, new wholes are created bridging the join between the assembly which are not themselves assembly components. When the carburettor is attached to the engine a new contiguous whole is created consisting of the carburettor body and the inlet manifold, but since this is not manipulated as a whole during assembly it is not an assembly component. …
>
> A second kind of component is one which fulfils a single purpose, office or function in the working of the artifact. We call these functional components. Another and sometimes better term for them in the case of complex artifacts is subsystems. For example, the braking system on the car is a functional component, the transmission is another. Sometimes functional components are assembly components, for example the carburettor. Other are not, for example the braking system, consisting of brake pedal, brake pipes, servo system, hydraulic fluid reservoir, cylinders, plungers, valves, brake pads and disks etc. is so relatively filigree and so spread around the vehicle that it would be physically impossible to mount it into the car as an assembly component: it comes into being as the car is assembled, and only then can it work. (pp. 263-4)

Business objects

(Partridge, Business objects: re-engineering for re-use, 1996) not only makes the distinction but gives an extensional analysis of the distinction – illustrating it with a space-time map.

> **2.3.1 Components as fusions of states**
>
> It is a truism that a whole is the sum of its parts. So it would seem reasonable to expect a thing to be the sum (the fusion) of all its components. However, object semantics reveals an inherent ambiguity in such everyday talk of components. At any point in time, it seems quite clear what a thing's components are. But it becomes much less clear when we consider different points in time.
>
> Here is an example that illustrates the problem. We expect some of a car's components to change. For instance, it is customary to change a car's tyres when they are worn; it is illegal not to. When we change a car's tyre, it stays the same car. It still has its full complement of components. It is just that one of its components has been changed. But what is this component we are talking about? It is one tyre before the change and another tyre after the change.

Object semantics gives a clearer and more accurate answer. Look at the space-time map of the car object #20 in ***Figure 8.14***. This shows that the four-dimensional extension of the car contains a temporal part of one tyre (#21) followed by the temporal part of another tyre (#22). At any one time, the car overlaps with only one tyre; but, over time, it overlaps with two tyres. (You may recognise this as a similar pattern to the chairman thought experiment.) The two tyres have state objects that are ‘components’ of the car.



If this is as far as we go, then the car could be said to have a different component before and after the change. But this is not at all satisfactory, because it would mean that the ‘components’ change over time—an anathema in our timeless object paradigm. We need a timeless explanation. We get it by constructing a tyre component from the fusion of all the ‘component’ tyre states. This is shown in the space-time map as the car’s tyre component—object #25. It is a component of the car; it is a part of the car; it is a fusion of the tyre state objects (#’s 23 & 24); and most important of all, it is timeless. This more sophisticated object-oriented component has none of the inherent ambiguity of our everyday notion.

Things and their parts

(Fine, “Things and their parts”, 1999) provides a very different analysis of the distinction which he calls rigid (integral) and variable (replaceable) embodiments as part of a radically new framework for material objects.

> Central to the paper is a distinction between two different ways in which one thing can be part of another. It can, in the first place, be a part in a way that is relative to a time. It is in this way, for example, that a newly installed carburetor is now a part of my car, whereas earlier it was not, or that certain molecules are now parts of my body though later, through the exercise of natural bodily functions, they no longer will be.
>
> In the second place, one object can be a part of another in a way that is not relative to a time. For something that is a part in this way, it is not appropriate to ask when, or for

how long, it is a part; it just is a part. It is in such a way that the pants and the jacket, for example, are parts of a suit or various atoms are parts of a water molecule, or two particular pints of milk are parts of a quart of milk, or various time-slices, if there are such things, are parts of a persisting individual.

It is by attempting to understand the relations of temporary and timeless part that we hope to come to a better understanding of material things. ... I then propose an alternative account that I believe is immune from the difficulties in the standard conceptions. This account takes seriously the idea that there is both a formal and material aspect to most material things. Thus it falls squarely within the hylomorphic tradition of Aristotle.

The particular version of hylomorphism developed here I call the theory of embodiment. It is in two main parts: the first, the theory of rigid embodiment, provides an account of the timeless relation of part (section3); and the second, the theory of variable embodiment, provides an account of the temporary relation (section 4). This theory requires us to accept two new kinds of whole and, in addition to its mereological consequences, has significant consequences for the nature of material things, their existence, their classification into sorts, and their interconnection with abstract objects ... (pp. 61-2)

What is pump facility PF101? A study in ontology

(Partridge, What is Pump Facility PF101? A Study in Ontology, 2002) builds upon (Partridge, Business objects: re-engineering for re-use, 1996) and provides a very detailed examination of the difference between integral and replaceable parts for the specific case of Pump PF101.

..., it shows how the simple general patterns that lurk in familiar places can be extremely difficult to extract without an ontological framework. The case study provides a number of examples. One simple one is that it was (and is) intuitively obvious to the EPISTLE engineers that (what ECM v2.22 terms) pump facilities and physical pumps were both pumps. However, the general distinctions made in the ECM v2.22 framework (which generalised other important insights) meant that this simple insight could not be accommodated. The ontological resources of the BORO framework resolve this issue.

Thirdly, it reveals a general pattern that is commonplace in business models – one that it extremely likely to be re-used. Pump facility can be seen as an example of a type of role that persists through a change of its occupier – and so is, in some sense, independent of it. Another classic example is the way organisation roles – such as Chairman of the Board – are independent of the specific person occupying the role. Though in both cases they are dependent of having a pump/person of some type to 'play' the role. Clearly it is useful to have a general pattern for these types of situation that can consistently and intuitively be applied across the board.

Replaceable parts: A four-dimensional analysis

(West, "Replaceable parts: A four dimensional analysis", 2003) returns to the topic, but this time builds upon the 4-dimensional analysis in (Partridge, What is Pump Facility PF101? A Study in Ontology, 2002)

Introduction

Replaceable parts (also known as facilities, tag parts, components or functional components) is a concept that is relevant, in particular, to many complex artefacts. These include general engineering products such as refineries and aircraft, and artefacts of the built environment, such as roads, buildings and bridges.

Little work is found on the subject of replaceable parts. Even Simons [1] only mentions component parts briefly, although there is additional material in [2]. However, Partridge [3] gives a detailed analysis of a particular case from the Process Industry, and this brief paper draws on this work. This brief paper introduces the concept of replaceable parts by looking at examples, examining their nature in intuitive terms, and finally gives an analysis of the concept in four- dimensional terms.

[1] Simons, Peter M. Parts a study in ontology Oxford University Press, 1987.

[2] Simons, Peter M. Charles W. Dement, Charles W. Aspects of the Mereology of Artifacts pp 255-276 in Roberto Poli and Peter Simons, eds., Formal Ontology, Dordrecht: Kluwer, 1996.

[3] Partridge, Chris. What is pump facility PF101? LADSEB-CNR – Technical report 04/02, 2002.

Developing high quality data models – 2011

(West, Developing high quality data models, 2011) continues the analysis raised in (West, The Data Management Guide: Developing High Quality Data Models: Volume 3, 1994) and (West, “Replaceable parts: A four dimensional analysis”, 2003)

14.2 The nature of system components

From the examples above a number of points arise:

- There are three objects involved, the system, the system component, and the item that is installed.
- The system component can be completely replaced in a material sense and its identity survives.
- The system component can survive periods when no item is installed, i.e. its existence is not necessarily continuous.
- The system component is coincident with the item installed whilst it is installed.
- The system component does not survive the destruction of the object it is a part of, though an item installed may do.
- Where there are multiple levels of assembly there may be a system component for each level, where each is existence dependent on the level it is part of. ...
- Figure 14 2 shows a space-time map for a system component. Here, Tag P101, the system component, undergoes but survives complete replacement. What can be seen is that the system component consists of the temporal parts of the ordinary physical objects that are installed.

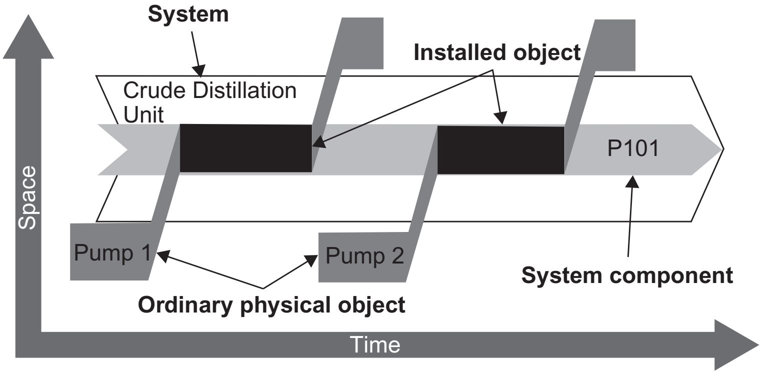


Figure 14-2 A space-time map for a system component.

Appendix B – Some literature on modularity and components

There is a significant literature on modularity and components across multiple disciplines. While different disciplines emphasise different aspects, they have a common core.
The coverage across disciplines is uneven, where some (such as computer science, complex systems and philosophy of mind) take a greater explicit interest than others – for example mathematics.

We give a very brief and selective, maybe eclectic, overview of this literature here – sufficient to give the impression that it is an important notion across a wide range of disciplines. For those interested in further reading, the next section contains two texts with general literature reviews, later texts also have useful references.

Cross discipline

(Campagnolo et al., "The concept of modularity in management studies: a literature review", 2010) gives a good overview of the literature. They make the excellent observation that the aim of modularisation affects (and is affected by) other elements:

Therefore, according to our point of view, a strong causality exists between the aim of modularization, definition of modules, modularization method and modularity measures, … which highlights that the choice of modularization objective affects and is affected by some other product, market and industry characteristics.")

(Schilling, "Modularity in multiple disciplines", 2002) contains the following table (Table 1) that gives some idea of the range of disciplines and uses of modularity. This table uses the author's selection of disciplines (which seem a little arbitrary) and categories of usage; there are many other categorisations, some more useful than others. We have not seen a need in our work to assess these – the categorisation here is provided as an illustration.

Concept	Psychology	Biology	American Studies	Mathematics	Technology and Organisations
Domain specific	X	X			X
Innately specified	X	X			
Hierarchically nested	X	X	X	X	X

More internal integration than external integration (localized processes and autonomy)	X	X	X	?	X
Informationally encapsulated	X		?		X
Near decomposability (segmentability, delineation between modules, or breaking down complexity)	X	X	X	X	X
Substitutability	X	?	X	?	X
Recombinability		X	X		X
Expandability		X	X	X	X
Module as homologue		X	X	X	X
Modules may be different in kind	X	X	X		X
Evolution from greater integration to greater modularity	X				X

Table 1: The use of modularity by discipline

Within discipline

The two disciplines probably most relevant to computer ontology development are computer science and mathematics. The first because computer ontologies are a computer science artefact – and computer science has thought extensively about how to produce modular artefacts. The second because the logical formalisation of computer ontologies is, from one perspective, mathematical. These two disciplines are covered first. Then, as a computer ontology of modularised engineering artefacts, would need to be able to represent these modules, we include a brief section in engineering modularisation.

In computer science

In computer science the interest in modularity and components is typically as a mechanism in the design; where the boundaries of the components are built into the artefact. In the early stages of programming, as evidenced by (Parnas, “On the criteria to be used in decomposing systems into modules”, 1972), the focus of effort was breaking programs into modules. In the seminal paper, (Dijkstra, “On the role of scientific thought”, 1974), provided an approach to designing the breakdown; the separation of concerns.

> But nothing is gained — on the contrary! — by tackling these various aspects simultaneously. It is what I sometimes have called "**the separation of concerns**", which, even if not perfectly possible, is yet the only available technique for effective ordering of one's thoughts, that I know of. This is what I mean by "focusing one's attention upon some aspect": it does not mean ignoring the other aspects, it is just doing justice to the fact that from this aspect's point of view, the other is irrelevant. It is being one- and multiple-track minded simultaneously.

There is now a vast literature on, among other things, Component Based Architecture and Component based Software Engineering – as well as frameworks and languages to facilitate the componentisation process. These practices then form the basis for more specific service-oriented or event driven architectures.

In computer ontology

These practices have 'leaked' into computer ontology development, see (Stuckenschmidt et al., Modular ontologies: concepts, theories and techniques for knowledge modularization, 2009), though this deals more with conceptual rather than foundational ontologies (see Appendix F).

In complex systems

(Simon, "The architecture of complexity", 1962) is a seminal paper in this area, which suggests modularity is ubiquitous, saying a complex system is:

> [r]oughly, ... one made up of a large number of parts that interact in a nonsimple way." It suggests that 'modularity' or 'nearly completely decomposable' is ubiquitous; a candidate for something "which, abstracting from properties peculiar to physical, biological, or social systems, would be applicable to all of them." Where modularity implies a hierarchical system, one "that is composed of interrelated sub- systems, each of the latter being, in turn, hierarchic in structure until we reach some lowest level of elementary subsystem." And these include "SYMBOLIC SYSTEMS ... systems of human symbolic production.

It notes the relation of decomposition to comprehensibility in the challenging case of complex systems saying "[t]he fact, then, that many complex systems have a nearly decomposable, hierarchic structure is a major facilitating factor enabling us to understand, to describe, and even to "see" such systems and their parts.

Another common theme is the evolution of modules as nature's way of dealing with complexity – see (Clune et al., "The evolutionary origins of modularity.", 2013).

These themes are developed for various disciplines in many places, including (Werner et al., Modularity: understanding the development and evolution of natural complex systems, 2005).

In cognitive science

Modular hierarchies have been used as to characterize the way the mind processes knowledge for some time. For example, in the seventeenth century, John Locke proposed a "mental chemistry" of the human mind, explaining how knowledge is built up hierarchically from simple ideas.

In the contemporary literature, (Fodor, The modularity of mind, 1983) is the classic text, that initiated the subsequent surge of interest. (Robbins, "Modularity of mind", 2009) is a good summary and critique of the current situation.

In mathematics

The process of breaking down has a long history in mathematic. A classic example is (Descartes, A Discourse on the Method, 2006) where it is the second of his four methods:

> In earlier years I had made some study of logic in the philosophy course, and of geometrical analysis and algebra in mathematics, three arts or branches of knowledge that seemed destined to contribute to my plan. … This was why I thought that another method had to be found which retained the advantages of all three but was free from their defects. …
>
> The second was to divide all the difficulties under examination into as many parts as possible, and as many as were required to solve them in the best way.

Hilbert's formalism led to a shift towards axiomatic mathematics, and this in turn raises the question of the organisation of these axioms, particularly in large systems. A classic text on this issue is (Bourbaki, "The Architecture of Mathematics", 1950). For a more modern text focusing on modularity see (Avigad, "Modularity in mathematics", 2020).

Graph theory provides a specific measure for modularity based upon the graph structure of networks – see, for example, (Brandes et al., "On Modularity Clustering", 2008). It measures the strength of division of a network into modules in terms of the density of the inter and intra module links – so a measure of clustering, where 'strong' modules have dense internal connections and sparse external connections.

Engineering

Of course, there is a strong interest in modularisation in engineering – see (Brusoni et al., "Patterns of modularization: The dynamics of product architecture in complex systems", 2011).

Appendix C – A brief etymological analysis

As one might expect, the English words associated with composition have a rich history – involving notions of putting together to make a whole and having distinct parts – reaching back to medieval times. This shows how the formal structures we describe in the body of the report emerge from deep within our informal linguistic heritage.

Term	Part of Speech	Etymology
assemble	verb	early 14c., transitive ("collect into one place") and intransitive ("meet or come together"), from Old French assembler "come together, join, unite; gather" (11c.), ... Meaning "to put parts together" in manufacturing is from 1852. Related: Assembled; assembling. ...
assembly	noun	... Meaning "a gathering together" is recorded from early 15c.; that of "act of assembling parts or objects" is from 1914, as is assembly line.
component	noun	1640s, "constituent part or element" ... from Latin ... present participle of *componere* "to put together, to collect a whole from several parts," ...
compose	verb	... Meaning "to make or form by uniting two or more things" is from late 15c. Sense of "be the substance or elements of, make up" is from 1540s. ... from 1650s as "place (parts or elements) in proper form, arrange.
composite	adjective	"made up of distinct parts or elements," c. 1400, from Old French composite, from Latin compositus "placed together," past participle of componere "to put together, to collect a whole from several parts," from com "with, together" (see com-) + ponere "to place" (past participle positus; see position (n.)). ...
composition	noun	late 14c., composicioun, "action of combining," also "manner in which a thing is composed," from Old French composicion (13c., Modern French composition) "composition, make-up, literary work, agreement, settlement," and directly from Latin compositionem (nominative compositio) "a putting together, connecting, arranging," noun of action from past participle stem of componere "to put together, to collect a whole from several parts," from com "with, together" (see com-) + ponere "to place" (past participle positus; see position (n.)).

compound	verb	late 14c., compounen, “to put together, to mix, to combine; to join, couple together,” from Old French compondre, componre “arrange, direct,” and directly from Latin componere “to put together,” from com “with, together” (see com-) + ponere “to place” (see position (n.)). The unetymological -d appeared 1500s in English by the same process that yielded expound, propound, etc. Intransitive sense is from 1727. Related: Compounded; compounding.
compound	noun	“a compound thing, something produced by the combination of two or more ingredients,” mid-15c., from compound (adj.).
compound	adj	late 14c., originally compounded, “composed of two or more elements, mixed, blended,” past participle of compounen (see compound (v.)). Of flowers from 1660s; compound eye is attested from 1836; compound sentence, one consisting of two or more full clauses, is from 1772.
constituent	noun	The notion is “to make up or compose” a body by appointing or electing a representative. ... . Meaning “that which constitutes as a necessary part, a formative element” is from 1756.
participate	verb	1530s, “to partake, to share or share in,” a back-formation from participation, or else from Latin participatus, past participle of participare “to share, share in, participate in; to impart,” from particeps “partaking, sharing,” from parti, past participle of partir “to divide” (from Latin partire, from pars “a part, piece,” ...).
participation	noun	“act or fact of sharing or partaking in common with another or others; act or state of receiving or having a part of something,” late 14c., participacioun, from Old French participacion (13c.) and directly from Late Latin participationem (nominative participatio) “partaking,” ... also, as an adjective, “sharing, partaking,” from pars (genitive partis) “a part, piece, a division” ...

Details retrieved from https://www.etymonline.com

Appendix D – A brief history of composition and whole-ness

The idea that the whole is the sum of its component parts runs deep. It can be traced back through the history of Western civilization – as the extracts below show.

> Because, if a thing has parts, the whole thing must be the same as all the parts. (Plato, Theaetetus, 1961, sec. 204a)

For more detail on Plato see (Harte, Plato on parts and wholes: The metaphysics of structure, 2002)

> Every individual thing is identical with its separate parts connected into one thing. For example, a man is identical with his head, chest, abdomen, feet, and other parts conjoined. (Boethius, In Ciceronis Topica, 1988, sec. i, 2.8)

> Whoever attributes existence to this house surely concedes the same to this stone and all the other parts taken together; for this house is nothing other than this stone together with the other parts. (Abelard, Dialectica, 1956, sec. v, i, 4)

> The parts taken together differ only in name from the whole. (Leibniz, “Specimen geometriae luciferae”, 1849, p. 274)

> A composite is nothing else than a collection or aggregatum of simple substances. (Leibniz, “The Monadology”, 1960, p. 455)

> Allness, or totality, is nothing other than plurality considered as a unity. (Kant, Critique of pure reason, 1934, sec. B 111)

> The idea is that if there are six things then a seventh thing – the whole they are parts of – exists, and the six things collectively are the seventh thing. (Baxter, “Identity in the loose and popular sense”, 1988, p. 579)

More recently, in (Yi, “Is mereology ontologically innocent?”, 1999), a distinction has been made between the strong claim that the whole is identical with the fusion of the parts and the weaker claim that fusion ‘a lot like identity’ claim. This creates space for people who wish to work with non-classical mereologies.

When developing a framework for composition, it is helpful if one is clear whether one is working with classical mereology – and if not, what one’s position with regard to composition as identity is.

Appendix E – Logical formulae

This appendix collects the formalisation included in the body of the report, providing a single point of reference. As noted in the paper, the goal of formalisation is to provide clarity in places, not to stand as a full-fledged axiomatisation. As part of the consistency checking process this appendix will be used as the basis for the development of the formalisation into a reasonable input to a computerised translation into the Common Logic Interchange Format (CLIF) defined in ISO/IEC 24707:2007 -- Annex A. (ISO, 2007).

Formulae used in the main body

We make the fundamental propositions a little more rigorous by formalising them (in first-order logic with identity). Our aim is not to provide a complete theory, only to delineate for the sake of clarity. We introduce vocabulary we need, or just find useful, in formulating these propositions as we go. We make no attempt to provide a complete or minimal formalisation and we leave open whether it could be replaced or condensed by an alternative axiomatisation.

We collect all the formalisation in the report into one place, for ease of reference, in this Appendix. Where it makes sense, we have also included the formalisation in the body of the text, highlighted in grey (different margins/font). Otherwise, we include this in a section at the end of this appendix.

We refer here to a standard account of the classical mereological fusion operator that allows us to denote the sum of all objects that satisfy a given property, ϕ. This is standardly defined in terms of the Parthood relation. Under these assumptions, given a property, the fusion of the objects satisfying this property exists and is unique. This implies a sophisticated formal apparatus that we will use here for the convenience of presentation but recognise that we may need to consider an alternative treatment in a fuller formalisation.

Symbols and conventions

$\vee$ for disjunction

$\wedge$ for conjunction

$\neg$ for negation

$\rightarrow$ for material implication

$\equiv$ for material equivalence (We do not use a definitional variant)

$\exists$ for the existential quantifier

$\forall$ for the universal quantifier

Vocabulary

(Elements)

Element(x) means x is an element.

wholePart(x,y) means that y is a part of x, where x and y are elements.

disjointFrom(x,y) means that x and y are mereologically disjoint, where x and y are elements, that is to say, according to standard mereological definitions, they do not share a part.

σxϕ(x) denotes the fusion of the x's such that ϕ(x), where the x's are elements.

PartRelation(x) means that x is a particular part relation between elements. We use the predicates partInRelation and wholeInRelation to write 'partInRelation(y,x)' (respectively, 'wholeInRelation(z,x)') for y (respectively, z) is the part (respectively, the whole) in the part relation x, where y and z are elements.

(*Atomic compositions*)

AtomicComposition(x) means x is an atomic composition.

isAtomicComponentOf(x,y) means x is an atomic component of y, where y is an atomic composition.

isAtomicCompositeOf(x,y) means x is an atomic composite of y, where y is an atomic composition.

StrictAtomicComposition(x) means that x is a strict atomic composition.

dissectingPartInAtomicComposition(x,y) means x is an atomic composition for which y is one of the dissecting part relations, i.e., where y is a part relation instance and the whole in y is the atomic composite of x and the part in y is an atomic component of x

(*Hierarchies*)

atop(x,y) means that atomic composition x is atop atomic composition y.

atopAt(x,y,z) means that atomic composition x is atop atomic composition y and z is a component of x and the composite of y.

atomicJoinedInto(x,y,z) means that (the composite of) atomic composition x and (a component of) atomic composition y are joined into (doubleton) composition z.

isAtomicConstituentOf(x,y) means that x is an atomic constituent of (the composition) y.

isCompositeOf(x,y) means that x is the composite element of (the composition) y.

isFreeComponentOf(x,y) means that x is a free component of (the composition) y.

isJoinedElementOf(x,y) means that x is a joined element of (the composition) y.

isTopAtomicConstituentOf(x,y) means that atomic composition x is a top atomic constituent of (the composition) y.

isOpenAtomicConstituentOf(x,y) means that atomic composition x is an open atomic constituent of (the composition) y.

isClosedAtomicConstituentOf(x,y) means that atomic composition x is a closed atomic constituent of (the composition) y.

MolecularComposition(x) means x is a molecular composition.

Composition(x) means x is a composition.

PairwiseNoPartAtomicComposition(x) means that atomic composition x satisfies the relaxed ‘no parthood’ constraint, i.e., no two components of x are in the whole-part relation.

PairwiseNoPartComposition(x) means that composition x respects ‘no parthood’, i.e., all its atomic constituents do.

topJoinedInto(x,z,w) means that the composition x is joined at the top to atomic composition z to build composition w.

bottomJoinedInto(z,x,w) means that atomic composition z is joined to the bottom of composition x to build composition w.

Propositions

(P1) $\forall x\, \forall y\, ((\text{isAtomicComponentOf}(x,y) \lor \text{isAtomicCompositeOf}(x,y)) \rightarrow \text{AtomicComposition}(y))$

(P2) $\forall x\, \forall y\, (\text{isAtomicComponentOf}(x,y) \rightarrow \neg\, \text{isAtomicCompositeOf}(x,y))$

(P3) $\forall x\, \forall y\, (\text{AtomicComposition}(x) \rightarrow (\text{isAtomicCompositeOf}(y,x) \equiv (y = \sigma z(\text{isAtomicComponentOf}(z,x)))))$

(P4) $\forall x\, \forall y\, \forall z\, ((\text{isAtomicCompositeOf}(x,y) \land \text{isAtomicCompositeOf}(z,y)) \rightarrow x=z)$ (Unicity of Atomic Composite)

(P5) $\forall x\, (\text{AtomicComposition}(x) \rightarrow \exists y\, \exists z\, (\text{isAtomicComponentOf}(y,x) \land \text{isAtomicComponentOf}(z,x) \land \neg\, (y=z)))$ (Multiplicity of Atomic Components)

(P6) $\forall x\, \forall y\, ((\text{AtomicComposition}(x) \land \text{AtomicComposition}(y) \land \forall z\, (\text{isAtomicComponentOf}(z,x) \equiv \text{isAtomicComponentOf}(z,y))) \rightarrow x=y)$ (Atomic Composition Criterion of Identity)

(P7) $\forall x\, \forall y\, \forall z\, ((\text{PartRelation}(x) \land \text{partInRelation}(y,x) \land \text{wholeInRelation}(z,x)) \rightarrow \text{wholePart}(z,y))$

(P8) $\forall x\, \forall y\, \forall u\, \forall v\, ((\text{PartRelation}(x) \land \text{PartRelation}(y) \land \text{partInRelation}(u,x) \land \text{partInRelation}(u,y) \land \text{wholeInRelation}(v,x) \land \text{wholeInRelation}(v,y)) \rightarrow x=y)$ (Part Relation Criterion of Identity)

(P9) $\forall x\, \forall y\, ((\text{AtomicComposition}(x) \land \text{AtomicComposition}(y) \land \forall z\, (\text{dissectingPartInAtomicComposition}(x,z) \equiv \text{dissectingPartInAtomicComposition}(y,z))) \rightarrow x=y)$

(P10) $\forall x\, \forall y\, \forall z\, ((\text{AtomicComposition}(x) \land \text{isAtomicComponentOf}(y,x) \land \text{isAtomicCompositeOf}(z,x)) \equiv \exists r\, (\text{PartRelation}(r) \land \text{dissectingPartInAtomicComposition}(x,r) \land \text{partInRelation}(y,r) \land \text{wholeInRelation}(z,r)))$

(P11) $\forall x\, \forall y\, \forall z\, ((\text{isAtomicComponentOf}(y,x) \land \text{isAtomicCompositeOf}(z,x)) \rightarrow \text{wholePart}(z,y))$

(P12) $\forall x\, (\text{StrictAtomicComposition}(x) \equiv (\text{AtomicComposition}(x) \land \forall y\, \forall z\, ((\text{isAtomicComponentOf}(y,x) \land \text{isAtomicCompositeOf}(z,x) \land \neg\, (y=z)) \rightarrow \text{disjointFrom}(z,y))))$

(P13) $\forall x\, \forall y\, \forall z\, ((\text{isAtomicComponentOf}(x,y) \land \text{isAtomicComponentOf}(z,y) \land \neg\, (x=y)) \rightarrow \text{disjointFrom}(x,z))$

(P14) $\forall x\, \forall y\, (\text{atop}(x,y) \equiv (\text{AtomicComposition}(x) \land \text{AtomicComposition}(y) \land \exists z\, (\text{isAtomicComponentOf}(z,x) \land \text{isAtomicCompositeOf}(z,y))))$

(P15) $\forall x\, \forall y\, (\text{atop}(x,y) \rightarrow \exists z\, (\text{MolecularComposition}(z)))$

(P16) $\forall x\, \forall y\, (\text{atop}(x,y) \rightarrow \exists z\, (\text{MolecularComposition}(z) \land \text{atomicJoinedInto}(x,y,z)))$

(P17) $\forall x\, \forall y\, \forall z\, (\text{atomicJoinedInto}(x,y,z) \rightarrow (\text{isAtomicConstituentOf}(x,z) \land \text{isAtomicConstituentOf}(y,z)))$

(P18) ∀x ∀y (isAtomicConstituentOf(x,y) ≡ ∃z (atomicJoinedInto(x,z,y) ∨ atomicJoinedInto(z,x,y)))

(P19) ∀x ∀y ∀z ((MolecularComposition(x) ∧ isTopAtomicConstituentOf(y,x) ∧ isTopAtomicConstituentOf(z,x)) → y = z)

(P20) ∀x ∀y ∀z ((MolecularComposition(x) ∧ isCompositeOf(y,x) ∧ isCompositeOf(z,x)) → y = z)

(P21) ∀x ∀y ∀z ((isCompositeOf(x,y) ∧ isAtomicConstituentOf(x,z)) → isTopAtomicConstituentOf(z,y))

(P22) ∀x ∀y ∀z ((isCompositeOf(x,y) ∧ isAtomicConstituentOf(y,z) ∧ isAtomicConstituentOf(u,z)) → u=y)

(P23) ∀x ∀y (isFreeComponentOf(x,y) → ∃z (isAtomicConstituentOf(z,y) ∧ isAtomicComponentOf(x,z)))

(P24) ∀x ∀y ∀u ∀v ((isFreeComponentOf(x,y) ∧ isAtomicConstituentOf(u,y) ∧ isAtomicConstituentOf(v,y) ∧ isAtomicComponentOf(x,u) ∧ isAtomicComponentOf(x,v)) → u = v)

(P25) ∀x ∀y ∀z ((isFreeComponentOf(x,y) ∧ isAtomicConstituentOf(z,y) ∧ isAtomicComponentOf(x,z)) → isOpenAtomicConstituentOf(z,y))

(P26) ∀x ∀y (isJoinedElementOf(x,y) ≡ ∃u ∃v (isAtomicConstituentOf(u,y) ∧ isAtomicConstituentOf(v,y) ∧ ¬ (u = v) ∧ isAtomicCompositeOf(x,u) ∧ isAtomicComponentOf(x,v)))

(P27) ∀x ∀y (isOpenAtomicConstituentOf(x,y) ≡ (isAtomicConstituentOf(x,y) ∧ ∃z isFreeComponentOf(z,x)))

(P28) ∀x ∀y (isClosedAtomicConstituentOf(x,y) ≡ (isAtomicConstituentOf(x,y) ∧ ¬ isOpenAtomicConstituentOf(x,y)))

(P29) ∀x ∀y ∀z ((Composition(x) ∧ isTopAtomicConstituentOf(y,x) ∧ AtomicComposition(z) ∧ atop(z,y)) → ∃w (topJoinedInto(x,z,w) ∧ ∀u (isAtomicConstituentOf(u,x) → isAtomicConstituentOf (u,w)) ∧ isAtomicConstituentOf(z,w)))

(P30) ∀x ∀y ∀z ∀w ((Composition(x) ∧ isTopAtomicConstituentOf(y,x) ∧ AtomicComposition(z) ∧ atop(z,y) ∧ topJoinedInto(x,z,w)) → isTopAtomicConstituentOf(z,w))

(P31) ∀x ∀y ∀z ((Composition(x) ∧ isOpenAtomicConstituentOf(y,x) ∧ AtomicComposition(z) ∧ atop(y,z)) → ∃w (bottomJoinedInto(z,x,w) ∧ ∀u (isAtomicConstituentOf(u,x) → isAtomicConstituentOf (u,w)) ∧ isAtomicConstituentOf(z,w)))

(P32) ∀x ∀y ∀z ∀w ∀u ((Composition(x) ∧ isOpenAtomicConstituentOf(y,x) ∧ AtomicComposition(z) ∧ atop(y,z) ∧ bottomJoinedInto(z,x,w) ∧ isTopAtomicConstituentOf(u,x)) → isTopAtomicConstituentOf(u,w))

(P33) ∀x ∀y ∀z ∀u (joinedInto(x,y,z) → (isAtomicConstituentOf(u,z) ≡ (isAtomicConstituentOf (u,x)) ∧ isAtomicConstituentOf(y,z))

(P34) Composition(x) ≡ (AtomicComposition(x) ∨ MolecularComposition(x))

(P35) ∀x ∀y ∀z ((Composition(x) ∧ Composition(y) ∧ ∀z (isAtomicConstituentOf(z,x) ≡ isAtomicConstituentOf (z,y))) → x=y) (Composition Criterion of Identity)

(P36) ∀x (PairwiseNoPartAtomicComposition(x) ≡ (AtomicComposition(x) ∧ ∀y ∀z ((isAtomicComponentOf(y,x) ∧ isAtomicCompositeOf(z,x) ∧ ¬ (y=z)) → ¬ wholePart(z,y)))) (No Parthood for Atomic Compositions)

(P37) ∀x (PairwiseNoPartComposition(x) ≡ (Composition(x) ∧ ∀y (isAtomicComponentOf(y,x) → PairwiseNoPartAtomicComposition(y)))) (No Parthood for Compositions)

Additional formulae

(Class subsumption relations)

We here introduce propositions intended to recapitulate the class subsumption hierarchy to facilitate the export to CLIF. Note that some of these are redundant given the formulae above.

Also, new predicates are used:

DoubletonComposition(x) means that x is a doubleton molecular composition, i.e., it has two atomic constituents.

StrictComposition(x) means that x is a composition, atomic or molecular, that is strict.

StrictMolecularComposition(x) means that x is a molecular composition that is strict, i.e., its atomic constituents are strict.

PairwiseNoPartMolecularComposition(x) means that x is a molecular composition that is also such that PairwiseNoPartComposition(x)

(P38) ∀x (AtomicComposition(x) → Composition(x))

(P39) ∀x (MolecularComposition(x) → Composition(x))

(P40) ∀x (StrictComposition(x) → Composition(x))

(P41) ∀x (StrictAtomicComposition(x) → StrictComposition(x))

(P42) ∀x (StrictAtomicComposition(x) → AtomicComposition(x))

(P43) ∀x (StrictMolecularComposition(x) → StrictComposition(x))

(P44) ∀x (StrictMolecularComposition(x) → MolecularComposition(x))

(P45) ∀x (PairwiseNoPartComposition(x) → Composition(x))

(P46) ∀x (PairwiseNoPartAtomicComposition(x) → PairwiseNoPartComposition(x))

(P47) ∀x (PairwiseNoPartAtomicComposition(x) → AtomicComposition(x))

(P48) ∀x (PairwiseNoPartMolecularComposition(x) → PairwiseNoPartComposition(x))

(P49) ∀x (PairwiseNoPartMolecularComposition(x) → MolecularComposition(x))

(P50) ∀x (DoubletonComposition(x) → MolecularComposition(x))

(Class disjointness relations)

(P51) ∀x (AtomicComposition(x) → ¬ MolecularComposition(x))

Appendix F – Foundational and conceptual ontology distinction

For our current purposes it is useful to make a distinction between two broad senses of the term 'ontology' used for data models; foundational and conceptualization. In practice, these are two ends of a range, rather than disjoint senses.

These senses have arisen to meet different goals:

- Foundational ontologies are used to add rigour to conceptual modelling, especially when there are data interoperability requirements (Guizzardi et al., 2004),
- Conceptualization ontologies have been developed over the past decades to assist with data modelling on the Web (Gruber, 1993).

Foundational ontologies

This sense takes a view of ontology as rooted in philosophy, where it deals with reality – what exists. Quine puts the problem of ontology simply:

"What is there?" […] "Everything" (Quine, 1948)

Philosophy has invested significant effort over thousands of years to make sense of the question and find helpful ways of guiding the answer. The result is a contemporary view that an ontology is "the set of things whose existence is acknowledged by a particular theory or system of thought." (Lowe, 1995)

This view is particularly relevant in the context of data interoperability as it grounds ontology in reality (i.e. "the things whose existence is acknowledged") rather than one's subjective conception of what constitutes the real world.

A data modelling ontology is foundational when it has these ontological foundations. These provide answers to ontological questions (such as those raised in (Partridge et al., 2020). The foundations typically manifest themselves in the data model as a high-level system of inter-related entities that aims to encompassing most if not all domains – what is often called an 'upper ontology'.

Formalising the foundational 'upper ontology' often requires higher levels of expressivity than are found in some computer languages. But, on the other hand, this level of expressivity may not be needed for some simple applications.

Conceptualization ontologies

This sense emerged in the early 1990s and has been defined in a number of very similar ways as a 'specification of a conceptualization':

- an "explicit specification of a conceptualization" (Gruber, 1993)
- a "formal specification of a shared conceptualization" (Borst, 1997)
- "a formal, explicit specification of a shared conceptualization" (Studer et al., 1998)

What is central is the notion that these ontologies encode some conceptual view. And this places no inherent commitment to dealing with philosophical ontological questions. There is also no constraint on scope, as the conceptualization driving the ontology can be general, domain specific and even only application specific. In addition, it is worth noting that semantic web languages built for conceptual ontologies are designed around computational efficiency, which comes at a price of expressivity. One result is that ontologies built using these languages are often simple class hierarchies with very limited axiomatisations.

It is now common to regard schemas encoded in standard languages for the semantic web (for example RDF(S) and OWL) as automatically ontologies in this sense. Though, of course, apart from concerns about the expressivity of the language, there is no reason why foundational ontologies could not also be written (encoded) in these languages.

References

Abelard, P. (1956). Dialectica. Assen, Van Gorcum.

Avigad, J. (2020). “Modularity in mathematics.” The Review of Symbolic Logic, 13(1), 47–79.

Baxter, D. L. M. (1988). “Identity in the loose and popular sense.” Mind, 97, 576–582.

Bittner, T. & Smith, B. (2001). “A taxonomy of granular partitions.” In International Conference on Spatial Information Theory (pp. 28–43).

Boethius. (1988). In Ciceronis Topica. Ithaca (NY): Cornell University Press.

Borst, W. (1997). Construction of Engineering Ontologies. Enschede, The Netherlands.

Bourbaki, N. (1950). “The Architecture of Mathematics.” American Mathematical Monthly, 57(4), 221–232.

Brandes, U., Delling, D., Gaertler, M., Gorke, R., Hoefer, M., Nikoloski, Z. & Wagner, D. (2008). “On Modularity Clustering.” Knowledge and Data Engineering, IEEE Transactions on, 20(2), 172 –188. https://doi.org/10.1109/TKDE.2007.190689

Brusoni, S. & Prencipe, A. (2011). “Patterns of modularization: The dynamics of product architecture in complex systems.” European Management Review, 8(2), 67–80.

Campagnolo, D. & Camuffo, A. (2010). “The concept of modularity in management studies: a literature review.” International Journal of Management Reviews, 12(3), 259–283.

Clarke, B. L. (1981). “A Calculus of Individuals based on ‘Connection.’” Notre Dame Journal of Formal Logic, 22(3), 204–218.

Clune, J., Mouret, J.-B. & Lipson, H. (2013). “The evolutionary origins of modularity.” Proceedings of the Royal Society B: Biological Sciences, 280(1755), 20122863. https://doi.org/10.1098/rspb.2012.2863

Cotnoir, A. J. & Varzi, A. C. (forthcoming). Mereology. Oxford University Press.

De Laguna, T. (1922). “Point, line, and surface, as sets of solids.” The Journal of Philosophy, 449–461.

Descartes, R. (2006). A Discourse on the Method. Oxford University Press.

Dijkstra, E. W. (1974). “On the role of scientific thought.”

Fine, K. (1999). “Things and their parts.” Midwest Studies in Philosophy, 23(1), 61–74.

Fine, K. (2010). “Towards a theory of part.” The Journal of Philosophy, 107(11), 559–589.

Fodor, J. A. (1983). The modularity of mind. MIT press.

Gruber, T. R. (1993). “A translation approach to portable ontologies.” Knowledge Acquisition, 5(2), 199–220.

Guizzardi, G. & Wagner, G. (2004). “A Unified Foundational Ontology and some Applications of it in Business Modeling.” In Grundspenkis, Janis and Kirikova, Marite (Ed.), CAiSE Workshops (pp. 129–143). Riga Technical University, Riga, Latvia.

Harte, V. (2002). Plato on parts and wholes: The metaphysics of structure. Oxford University Press on Demand.

Husserl, E. (1970). Logical Investigations. London: Routledge & Kegan Paul.

ISO. “ISO/IEC 24707:2007 Information technology - Common Logic (CL).” (International Organization for Standardization, Ed.) (2007).

Kant, I. (1934). Critique of pure reason (p. 483, 1 l.). London, New York, E. P. Dutton & Co., Inc.: J. M. Dent & Sons Ltd.

Leibniz, G. W. F. von. (1849). “Specimen geometriae luciferae.” In C. I. Gerhardt (Ed.), Leibnizens mathematische Schriften. Halle: H.W. Schmidt.

Leibniz, G. W. F. von. (1960). “The Monadology.” In The Rationalists. Doubleday.

Lowe, E. J. (1995). “Ontology.” The Oxford Companion to Philosophy, 634–635.

Parnas, D. L. (1972). “On the criteria to be used in decomposing systems into modules.” Communications of the ACM, 15(12), 1053–1058.

Partridge, C. (1996). Business objects: re-engineering for re-use. Butterworth-Heinemann.

Partridge, C. (2002). What is Pump Facility PF101? A Study in Ontology.

Partridge, C., de Cesare, S., Mitchell, A., Gailly, F. & Khan, M. (2017). “Developing an Ontological Sandbox: Investigating Multi-level Modelling’s Possible Metaphysical Structures.” In Loli Burgueno et al (Ed.), Proceedings of MODELS 2017 Satellite Event co-located with ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS 2017), Austin, TX, USA, September, 17, 2017 (Vol. 2019, pp. 226–234). CEUR-WS.org.

Partridge, C., de Cesare, S., Mitchell, A. & Odell, J. (2016). “Formalization of the classification pattern: survey of classification modeling in information systems engineering.” Software & Systems Modeling, 1–37.

Partridge, C., Mitchell, A., Cook, A., Sullivan, J. & West, M. (2020). A Survey of Top-Level Ontologies - to inform the ontological choices for a Foundation Data Model.

Partridge, C., Mitchell, A., Loneragan, M., Atkinson, H., de Cesare, S. & Khan, M. (2019). “Coordinate Systems: Level Ascending Ontological Options.” In 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C) (pp. 78–87). Retrieved from https://www.academia.edu/40354620

Partridge, C., Mitchell, A., Loneragan, M., Atkinson, H., de Cesare, S. & Khan, M. (forthcoming). “The Fantastic Combinations and Permutations of Co-ordinate Systems’ Characterising Options: The Game of Constructional Ontology.”

Plato. (1961). Theaetetus. Princeton University Press.

Quine, W. V. (1948). “On what there is.” The Review of Metaphysics, 2(5), 21–38.

Robbins, P. (2009). “Modularity of mind.”

Schaffer, J. (2010). “Monism: The priority of the whole.” The Philosophical Review, 119, 31–76.

Schilling, M. A. (2002). “Modularity in multiple disciplines.” In Managing in the modular age: Architectures, networks and organizations (pp. 203–214). Blackwell Publishers and Editors: Raghu Garud, R Langlois, A Kumaraswamy.

Simon, H. A. (1962). “The architecture of complexity.” In Proceedings of the American Philosophical Society, (Vol. 106, pp. 467–482). American Philosophical Society.

Simons, P. (1987). Parts: A Study in Ontology.

Simons, P. & Dement, C. W. (1996). “Aspects of the Mereology of Artifacts.” In Formal ontology (pp. 255–276). Springer.

Stein, S. & Szabó, S. (1994). Algebra and Tiling: Homomorphisms in the service of Geometry. The Mathematical Association of America.

Stuckenschmidt, H., Parent, C. & Spaccapietra, S. (2009). Modular ontologies: concepts, theories and techniques for knowledge modularization (Vol. 5445). Berlin Heidelberg: Springer-Verlag.

Studer, R., Benjamins, R. & Fensel, D. (1998). “Knowledge engineering: Principles and methods.” Data & Knowledge Engineering, 25(1-2), 161–198.

Uzquiano, G. (2011). “Mereological Harmony.” Oxford Studies in Metaphysics, 6, 199–224.

Werner, C., Rasskin-Gutman, D. & Simon, H. (2005). Modularity: understanding the development and evolution of natural complex systems. Cambridge, MA: MIT Press.

West, M. (1994). The Data Management Guide: Developing High Quality Data Models: Volume 3.

West, M. (2003). “Replaceable parts: A four dimensional analysis.” In COSIT-03-Workshop on fundamental issues in spatial and geographic ontologies. Ittingen, Switzerland.

West, M. (2011). Developing high quality data models. Elsevier.

Whitehead, A. N. (1916). “La théorie relationniste de l’espace.” Revue de Métaphysique et de Morale, 23(3), 423–454.

Whitehead, A. N. (1919). An enquiry concerning the principles of natural knowledge (1st ed.). Cambridge: Cambridge University Press.

Whitehead, A. N. (1920). The concept of nature. Cambridge: Cambridge University Press.

Whitehead, A. N. (1929). Process and Reality: An Essay in Cosmology. Cambridge: Cambridge University Press.

Wimsatt, W. C. & Wimsatt, W. K. (2007). Re-engineering philosophy for limited beings: Piecewise approximations to reality. Harvard University Press.

Yi, B.-U. (1999). “Is mereology ontologically innocent?” Philosophical Studies: An International Journal for Philosophy in the Analytic Tradition, 93(2), 141–160.

Acknowledgements

Authors

Chris Partridge
Andy Mitchell
Pierre Grenon

Contributors

Hayden Atkinson
Ian Bailey
Sergio de Cesare
Al Cook
Andrew Davis
Salvatore Florio
Pawel Garbacz
Mesbah Khan
Steven Kraines
David Leal
Alex Luck
Jan Sullivan
Liz Varga
Matthew West

Centre for Digital Built Britain

The National Digital Twin programme is funded by the University of Cambridge and the Department for Business, Energy and Industrial Strategy via InnovateUK, part of UK Research and Innovation. This research was enabled by the Construction Innovation Hub(the Hub). The Hub is funded by UK Research and Innovation through the Industrial Strategy Challenge Fund.

Partridge, C., Mitchell, A. and Grenon, P. (2021) A Framework for Composition: A Step Towards a Foundation for Assembly. doi.org/10.17863/CAM.66459

CONSTRUCTION INNOVATION HUB

Department for Business, Energy & Industrial Strategy

INDUSTRIAL STRATEGY

UK Research and Innovation