# 1 A Formal Description of Backbone

In this section we formally describe how adding component definitions, defined using resemblance and replacement, adjusts the existing compositional structure of an architecture. As dependencies between strata govern the order of application and hence the interplay between replacement and resemblance, we begin by elucidating the stratum concept.

The ability to evolve an existing architecture in a decentralized manner, using strata to group these component definitions, leads logically to the desire to merge independently developed strata back into a unified architecture. We describe the merging rules, showing that any resulting structural errors can be corrected by adding further component definitions.

## 1.1 Strata and Strata Dependencies

A stratum is a hierarchical module that owns and groups elements such as components and interfaces. Each stratum explicitly indicates its dependence on other strata.

**Definition** (Stratum): A stratum is represented as the structure

where indicates a single possible parent stratum giving rise to nesting, is the set of other strata that depends on, and represents the set of elements owned by *s*.

**Definition** (Strata visibility): We define to be the closure of the dependencies . This represents all strata (and hence owned elements) visible to elements owned by . We further define to be the closure of the strata set , including also.

**Constraint** (Strata non-circularity): Strata dependencies must form a graph.

**Definition** (Strata independence): Two strata share a common base, but are independent, if neither have visibility of the other via their dependencies but each has visibility of common strata. This allows us to model two strata developed in isolation in a possibly decentralized manner, which nevertheless build on and possibly evolve elements in a common set of base strata.

## 1.2 Elements

**Definition** (Element): An element is represented as the structure

where is the owning stratum, is the optional element that this definition replaces and is the set of elements being resembled. The element being replaced cannot be from the same stratum This accords with the use of stratum as a unit of ownership: if the owner of a stratum wishes to alter an element within that stratum, then they can do this directly via destructive editing rather than via replacement.

An element’s expanded structure is determined by applying deltas ( to the structure inherited from the elements it resembles, resulting in a set of constituents. For instance, the constituents of a component are port, part, connector and attribute. is the set of constituents added by this definition, the set of inherited constituents which are to be deleted, and is an injection indicating a subset of inherited constituents to be replaced by new constituents.

**Constraint** (Element visibility): An element may only resemble or replace elements owned by the strata set. In other words, an element has visibility of other elements in its owning strata, and all strata that it transitively depends upon.

**Constraint** (No direct reference of a replacement): No element may directly replace or resemble a definition that replaces another element.

When replacing or resembling, the designer should reference the original definitions rather than any replacement definitions. Replacements will instead be considered when determining the expanded structure of each element. This rule allows us to later adjust the dependency order of strata without invalidating relationships.

**Constraint** (Single replacement): A stratum cannot contain two elements which both replace the same element. Without this restriction, we could potentially have multiple replacements inside a single stratum, which could not be otherwise ordered.

## 1.3 Interplay Between Replacement and Resemblance

As previously indicated, a stratum can contain element definitions which replace elements in other strata. This can affect existing resemblance relationships, as an element which was previously being resembled may be replaced with another definition. As such, we need to re-determine the resemblance graph for the entire system anew from the perspective of each stratum. We call this the expanded resemblance graph. We next show a number of definitions, leading towards a description of this concept.

**Definition** (Scoped replacements): We define to be the set of all components replacing element , defined in the transitive closure of the set of strata. We further add to the result for convenience in further expressions.

**Definition** (Replacements for resembles): We can now determine all possible replacements for the elements that resembles, from the perspective of stratum .

Note that the first term, which deals with the case where the definition both resembles and replaces the same element, does not consider replacements in stratum . This avoids circularity, as otherwise the term would pick up itself. In other words, we avoid to pick up the previous definition of the element being replaced.

**Definition** (Expanded resemblance): To form the expanded resemblance graph, for element from the perspective of stratum , we start with the replacements for all resembled elements defined in the dependency closure of . We then remove any replacing elements which are superseded by replacements higher up in the strata dependency graph.

This function can be used to compute an expanded resemblance graph for an element, from a given stratum. This may result in a multi-headed graph, if a stratum depends on multiple, independent strata which both replace the same element.

We further define as the closure of the function.

**Constraint** (Non-circularity of expanded resemblance): All expanded graphs must be non-circular. The following definition constrains the permissible resemblance and replacement relationships that an element may enter into.

## 1.4 Applying Deltas to Form the Expanded Set of Constituents

Using the expanded resemblance graph, we can now combine the deltas to form each element’s expanded definition from a given stratum perspective.

**Definition** (Expanded constituents): The expanded form of element from perspective is a set of binary relations that map from constituents added () to constituents added or replaced ().

We now tie this to the definition of element in 1.2 via the expanded resemblance graph. There are two cases: (a) where there is a single top to the expanded resemblance graph, and we need to apply the deltas of the top element to the inherited constituents, and (b) where there are multiple tops, representing independent branches of the graph that must be merged, and we need to union all of the associated deltas.

=

Explain more about multi-headed e-resemblance.

**Definition** (Delta merge): Merge applies the constituent deltas in the expanded resemblance graph to form an expanded set.

The first term is the union of the expanded constituent relations from all of the elements. The next two terms reapply the deletion and replacement constituents. This is necessary as tops may represent two or more independent branches of the resemblance graph that are now being merged.

The last three terms apply the add, delete and replace constituents from the possible top of the expanded resemblance graph. This shows how a further definition, which brings together independent branches of a resemblance graph, can correct any structural errors that result. In other words, a replacement definition will be able to add, delete and replace any inherited structural elements to correct the merged definitions for that element.

Replace overriding delete?

**Definition** (Invalid element): An element is invalid from the perspective of stratum if any of its domain constituents () map onto more than one range constituent ( -- in other words if this is not a function.

Other errors: deletion in one branch when other side expects it. Incompatible replace. Replace overriding delete. Higher level errors between elements – e.g. delete ports expected by another.