# 1 A Formal Description of Backbone

In this section we formally describe how component definitions, defined using resemblance and replacement, can evolve the existing compositional structure of an architecture.

The ability to evolve an existing architecture in a decentralized manner, using strata to group and organize 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.

As dependencies between strata govern the order of application and hence the interplay between replacement and resemblance, we begin by elucidating the stratum concept.

## 1.1 Strata and Strata Dependencies

A stratum is a hierarchical module that owns and groups elements definitions such as components and interfaces. Each stratum must explicitly indicate 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*.

As we will shortly illustrate, strata dependencies are important as they are used to control which elements can be legally referred to by elements in a stratum.

**Definition** (Strata visibility): We define to be the closure of the dependencies . We further define the following to also include itself.

We wish to explicitly outlaw circular references between strata, thereby forcing the dependencies into a graph. This allows us to divide strata into those a given stratum depends on, those that depend on it, and those it has no visibility of.

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

We now describe the concept of independent strata, which allows us to model two or more strata developed in isolation in a possibly decentralized manner. Although independent strata cannot see each others’ definitions, they may build on and evolve elements in a common set of base strata.

**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.

Strata that are independent, but do not share a common base, cannot each evolve common elements and therefore cannot conflict when combined.

In the example of figure 5, MultiCarBridge and FairBridge are independent strata that build on the common SingleLaneBridge stratum. As it is not possible for each independent stratum to refer to each other’s definitions, it is possible for each to have been developed in a separate environment and then brought back into a common environment to be unified into a single architecture. The FairMultiLaneBridge stratum builds on these definitions, relying on this single view.

## 1.2 Elements

An element can be either a component or an interface. It is owned by a single stratum. Evolution occurs when an element is defined to replace another in a different stratum.

**Definition** (Element): An element is defined by the following 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 simplifies the description, and 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 are able to do this directly via destructive editing rather than via replacement.

As previously discussed, strata dependencies govern which other elements a new element may legally refer to.

**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 in all strata that its owner transitively depends upon.

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 a relation indicating a subset of inherited constituents to be replaced by new constituents.

In section 2.2, we see that the textual representation of the MULTISLB component maps easily onto the above definition as follows.

Note that all constituents are treated uniformly, even if the textual syntax of definitions does not always appear to bear this out. As a further example, consider that the Controller’ evolution in figure 4 maps onto the following definition.

When replacing or resembling, the system designer should always reference the original definitions rather than any definitions that replace others. 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. We express this as the following constraint.

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

Furthermore, 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. We express this via the following.

**Constraint** (Single replacement):

## 1.3 Interplay Between Replacement and Resemblance

A stratum can contain element definitions which replace elements in other strata. As we apply these replacements, this will affect existing resemblance relationships: 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 , so as to pick up the previous definition of any 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.

As an example, the expanded resemblance graph for Controller is simply Controller from the perspective of stratum SingleLaneBridge. From the perspective of FairBridge and also FairMultiLaneBridge however it is Controller’ resembles Controller , reflecting that a replacement has superseded the original definition.

**Constraint** (Non-circularity of expanded resemblance): All expanded resemblance 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, which takes replacements into account, 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 ().

This approach allows us to hold the original and replacing constituents. [Example]

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.

=

**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 may represent two or more independent branches of the expanded resemblance graph that are now being merged – any deletion or replacement from one side of the branch will not have been applied to the other side. Delta replacement is modelled using the standard relational operator , which replaces any existing relations on the left hand side with a possible new one on the right with the same domain constituent.

The last three terms apply the add, delete and replace constituents from the possible single top of the expanded resemblance graph. These terms allow a further definition, which brings together independent branches of a resemblance graph, to correct any structural errors that result from the merging. 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.

Note that a replacement in one branch of the resemblance graph will override a deletion in another branch. This ensures that a merge preserves as much information as possible.

## 1.5 Detecting Structural Errors

Structural errors can result when combining two or more independently developed strata into a single architecture. This can happen, for instance, when combining two independent strata which (incompatibly) replace the same element in the common base strata. So, although our approach allows the definitions from independent strata to always be combined, the result may have structural errors despite each stratum being correct in isolation.

At a simple level, structural conflicts occur when both sides replace the same constituent in the same common, base element. This is captured by the following property.

**Property** (Conflicting replacement): An element is in error from the perspective of stratum if its expanded constituent relations do not form a function. In other words, if any of the expanded domain constituents () map onto more than one range constituent (.

At this point, we have described the approach without reference to a given component model. Other errors are higher level in nature, and must be related back to the syntactic and semantic rules of a given component approach. For instance, one branch may delete a part of a component that another branch relies on. One branch may replace a part with one with different ports, and a merge may find that the other side of the branch erroneously connects to the ports expected on the older constituent. A full component model and associated rules is able to be constructed on our delta approach, allowing these types of errors to be detected.

## 1.6 Correcting Structural Errors in an Architectural Context

As explained previously, the last three terms of the definition in section 1.4 allow a subsequent replacement of an element to add, replace or delete constituents to fix that element’s structure. The replacement must be in a stratum which depends on the independent stratum which are being combined, ensuring it is the single top of the expanded resemblance graph for that element. In that case, the three delta types (add, delete, replace) map onto the deltas held in the definition and any required changes can be effected to the inherited structure.

[example of hierarchy, error and correction]

To address: Behavioural errors, examples