```
fmArch150 = FM(FM ( Arch : Ar1 Ar2; Ar1: (Ar3|Ar4); Ar2 : (Ar5|Ar6); )
fmPlugin = FM ( Plugin : (Pl1|Pl2|Pl3)+ ; Pl1 -> Pl2 ; )
fmFull = aggregate { fmArch fmPlugin } withMapping constraints (Ar3 -> Pl1 ; Pl2 -> Ar5;
renameFeature (root fmFull) as "FtAggregation"
// enforced architectural FM
fmArch = slice fmFull including fmArch.*

// let us play with the models
configs fmArchEnforced

// we know by construction that fmArch either a refactoring or a specialization of fmArc
cmp = compare fmArch150 fmArch
assert (cmp eq REFACTORING || cmp eq SPECIALIZATION)

// or, equivalently, that the difference (or complement) bewteen the set of configuratio
// fmArch and fmArch150 is empty
assert (not (isValid merge diff { fmArch fmArch150 } ))

// we now compute the difference bewteen the set of configurations of fmArch150 and fmAr
 // fmArch150Removal is an FM that compactly represents this difference
fmArch150Removal = merge diff { fmArch150 fmArch }
if (compare fmArch fmArch eq SPECIALIZATION) then
                println "configurations removed from fmArch150=" configs fmArch150missing
else
        // refactoring
        println "configurations of fmArch150 have not been modified"
end

```