```
--- substitute activity diagram elements from each mitigation template
    foreach partial mitigation pm_i^h ∈ CM^h {
        --- for each two elements to be substituted…
        foreach operation op_i ∈_t R |R ∈_t pm_i^h {
            foreach element to be removed el^old, el^new ∈_t op_i^h {
                if el^old, el^new ∈_t op_i^h are of type activity {
                --- insert new activity by computing new set of activities
                    A' = A^actD^fr ∪ el^new
                --- remove old activity by computing new set of activities
                    A^actD^hmr = A'\el^old
                }
                if el^old, el^new ∈_t op_i^h are of type pin {
                --- insert new pin by computing new set of pins
                    P' = P^actD^fra ∪ el^new
                --- remove pin by computing new set of pins
                    P^actD^hmr = P'\el^old
                }
                if el^old, el^new ∈_t op_i^h are of type control node {
                --- insert new control node by computing new set of control nodes
                    C' = C^actD^fr ∪ el^new
                --- remove control node by computing new set of control nodes
                    C^ad^hmr = C'\e^old
                }
                if el^old, el^new ∈_t op_i^h are of type activity edge {
                ---insert new activity edge by computing new set of activity edges
                --- make sure source and target of the activity edge exist
                   E' = E^ad^fr ∪ el^new|el^new = (src, m, tar): src, tar ∈ A^actD^hmr ∪ P^actD^hmr ∪ C^actD^hmr
                ---remove activity edge by computing new set of activity edges
                   E^ad^hmr = E'\el^old
                }
            --- find all activity edges that have el^old as source
            E'^actD^fr = E^actD^hir\e|∀e = (src, m, tar) ∈ E^actD^fr: src ∈_t el^old ∨ tar ∈_t el^old
            --- for every element that has el^old as source
            foreach (e^old = (src, m, tar) ∈_t E'^actD^fr|src = el^old) {
                --- make a new activity edge to connect to the new element
                --- with same message, guard, and target
               e^new = (src, m, tar)|m, tar ∈_t e^old ∧ src = el^new
                --- insert new activity edge into set of activity edges
                E' = E^ad^hmr ∪ e^new
                ---remove activity edge by computing new set of activity edges
                E^actD^hmr = E'\e^old
            }
            --- for every element that has el^old as target
            foreach (e^old = (src, m, tar) ∈ E'^ad|tar = el^old) {
                --- make a new activity edge to connect to the new element
                --- with same source and message
               e^new = (src, m, tar)|src, m ∈_t e^old ∧ tar = el^new
                --- insert new activity edge into set of activity edges
                E' = E^ad^hmr ∪ e^new
                ---remove activity edge by computing new set of activity edges
                E^ad^hmr = E'\e^old
                ---remove activity edge by computing new set of activity edges
                E^ad^hmr = E'\e^old
```

```
54                  }
55              }
56          }
```

Listing 4 Pseudo-Code of substitution operation $op^{substitute}$ of the QVTo Script q^{hmr}.