Algorithm 0.1: Cluster of Callouts(coList)

procedure Init Clusters(coList)
for $i \leftarrow 0$ to coList.size()
do clList.insert(coList[i])
return (clList)

procedure NotAdded(coList)
count $\leftarrow 0$
for $i \leftarrow 0$ to coList.size()
do if clList[i].added == false
then {count $\leftarrow$ count + 1}
return (count)

procedure Overlap(cluster1, cluster2)
overlap $\leftarrow$ overlaps(cluster1.callouts(), cluster2.callouts())
comment: overlaps is measured by overlap matching
return (overlap)

main
clList $\leftarrow$ initClusters(coList)
current_cluster_count $\leftarrow$ cluster_List.size()
while (1)
do for $i \leftarrow 0$ to clList.size()
do if clList[i].added == false
then {for $j \leftarrow i + 1$ to clList.size()
do if (cluster_List[j].added == false)
then {overlap $\leftarrow$ Overlap(clList[i], clList[j])
if overlap > 0
then {clList[i].insert(clList[j])
clList[j].added $\leftarrow$ true

current_cluster_count $\leftarrow$ NotAdded(cluster_List)
if (current_cluster_count == cluster_count)
then break
cluster_count $\leftarrow$ current_cluster_count