

Setting size of comment

SCENARIO 1

CmdInsertComment

random id beginning with "C"

node = displaymodel.AddComment(id, comment) ^①

shape = umlcanvas.createCommentShape(node)

DisplayModel

AddCommentNode(id, comment) ^②

t, l, w, h = random

node = CommentNode(id, t, l, w, h, comment) ^③

add to graph

return node

GraphNode

CommentNode

init

super

self.comment = comment

UmlCanvas

createCommentShape(self, node) ^④

shape = CommentShape(node.width, node.height) ^⑤

shape.setSize(node.width, node.height) ^⑥

- ① When creating fresh nodes, getting random position + random small size due to ①. The shape ^{then} mirrors the node size. ②
- ③ Suggest adding param of default width & height at ③. Actually ⑤ not necessary, just put the initial default values for a newly inserted comment at ① ~ not so random.

however when loading a comment from persistence.....

SCENARIO 2

GraphPersistence

Load()

if comment

node = graph.create-new-comment(id, x, y, width, height)

graph.AddNode(node)

self.

graph.node-from-persistence-str(node, data)

CmdFileLoadLibraryBase

load.model....and build shapes (the

via Graph.LoadGraphFromStrings()

calls persistence.Load()

umlcanvas.build-view()

Graph

UmlGraph

create-new-comment(id, t, l, w, h) ^⑦return CommentNode(id, t, l, w, h) ^⑧

node-from-persistence-str(node, data)

node.comment = data['comment'] # base64 decoded etc

UmlCanvas

~~build~~

build-view

delete existing shapes

for node in graph.nodes:

shape = self.createCommentShape(node)

self.displaymodel.classnameToShape[node.id] = shape

create shape edges for all the edges.

data is the persisted line as a dict, containing width, height, id, comment etc.

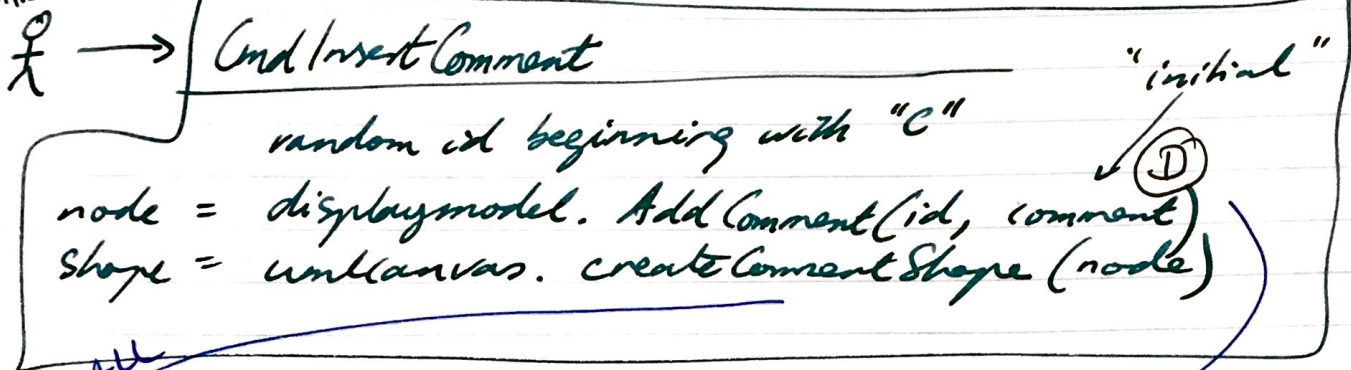
⑦ no random shift going on here, unlike ①

⑧ CommentNode not added to graph here, but in Load(). whereas in scenario 1, DisplayModel.AddCommentNode() creates node and adds it to graph. would be nice if the two scenarios worked more similarly, or similar functionality is being done in different places and in different chunks/ways

aha - does not delete existing edges! this is the source of another bug.

Setting size of comment &

SCENARIO 1



DisplayModel

AddCommentNode(id, comment) ②t, l, w, h = random ③node = CommentNode(id, t, l, w, h, comment) VI CREATEadd to graph
return node

GraphNode

CommentNode

init
super
self.comment = comment

ogl.ShapeCanvas

UmlCanvas

createCommentShape(self, node) CALL

④ shape = CommentShape(node.width, node.height)

⑤ shape.setSize(node.width, node.height)

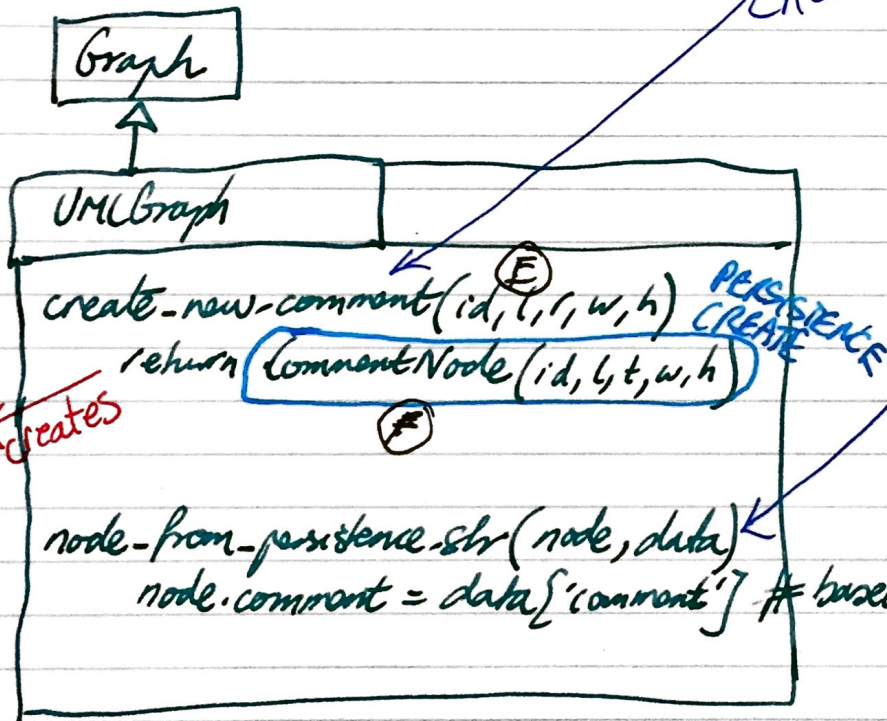
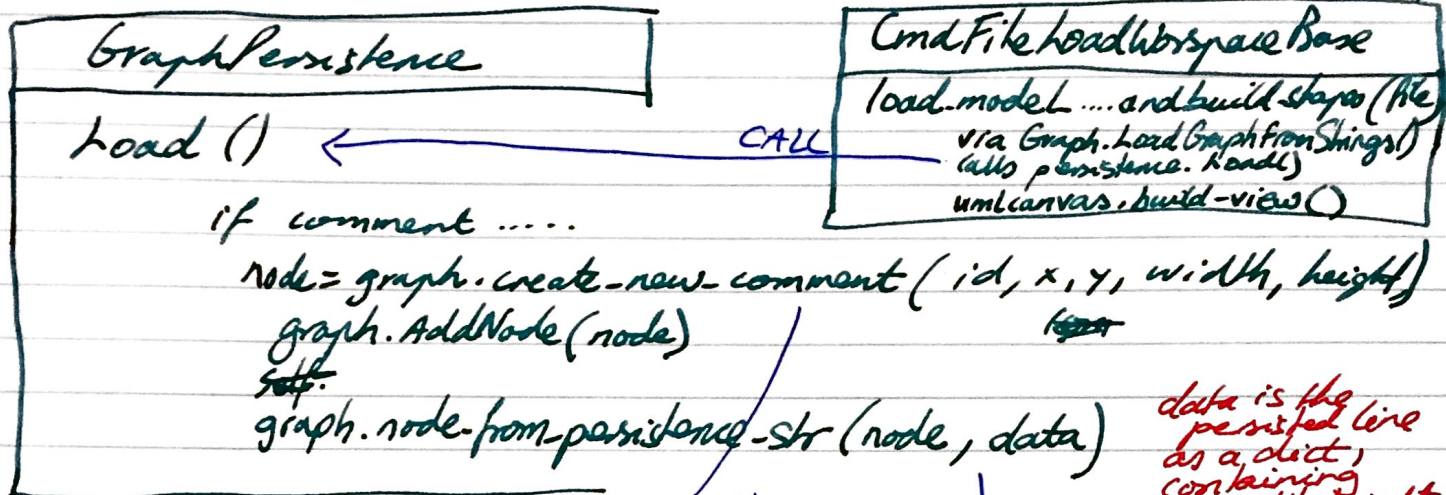
promising
prospect
for width
& height

why again?

- ① When creating fresh nodes, getting random position & random small size due to ①. The shape ^{than} mirrors the node size. ②
- ✓ Suggest adding param of default width & height at ③.
- Actually ③ not necessary, just put the initial default values for a newly inserted comment at ① ~ not so random.

SCENARIO 2

however when loading a comment from persistence.....



- (E) no random shift going on here, unlike (A)
- (F) comment node not added to graph here, but in Load(). whereas in scenario 1, DisplayModel.AddCommentNode() creates node and adds it to graph. would be nice if the two scenarios worked more similarly, cos similar functionality is being done in different places and in different chunks/ways

