Cognitive walkthrough: description and example

Based on Task-Centered Design by

Clayon Lewis & John Rieman, CU Boulder

Cognitive Walkthrough

- Formalized way of imagining people's thoughts and actions when they use an interface *for the first time.*
  - Not a technique for evaluating the system over time (e.g., how quickly a user moves from beginner to intermediate)
- First select a REAL, COMPLETE, REPRESENTATIVE TASK that the design is intended to support.
  - Task-Centered Design and Goal-Directed Design are both types of User-Centered Design
- Then try to tell a *believable story* about each action a user has to take to do the task.
  - IMPORTANT! Telling a believable story helps to stay "in persona" – filling in an Excel spreadsheet may not accomplish this same goal.
- To make the story believable, you have to motivate each of the user's actions, relying on the user's general knowledge and on the prompts and feedback provided by the interface. If you can't tell a believable story about an action, then you've located a problem with the interface.
  - Will the user try this action? May depend on background of user… make sure you get "in persona" before doing the walkthrough!
  - Is the action visible and easy to notice?
  - Is the action clear? Vocabulary problem (next slide)

What's it good for?

- Question assumptions about what the users will be thinking
- Identify controls that may be missing or hard to find (buried menus, issues with gesture-based interfaces)
- Note inadequate feedback (did I really make progress? How do I know?)
- Suggest difficulties with labels and prompts…. terms that are ambiguous or too techy. Multiple labels that *could* be the correct step.

The Vocabulary Problem

- Armchair naming. Designers use names that make sense to them.
- In a study to determine the likelihood that two people will apply the same name to an object:

Personnel	Domain	%
Typists	Describe text editing operation	11%
System Designers	Commands for a message decoder	8%
College students	First word used to describe common objects like love, motorcycle	12%
Expert cook/homemaker	Recipe keyword	18%

How to do it

Prior to doing a walkthrough, you need four things:

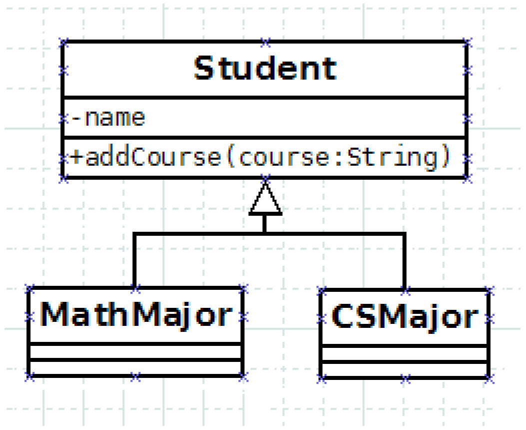
1. You need a description of a prototype of the interface. It doesn't have to be complete, but it should be fairly detailed. Things like exactly what words are in a menu can make a big difference.
2. You need a task description (for a representative task).
3. You need a complete, written list of the actions needed to complete the task.
4. You need an idea of who the users will be and what kind of experience they'll bring to the job.

Some caveats

- **Don't merge step 3 into the evaluation process.** The walkthrough should look at the exact sequence, to identify problems users might encounter when following it.
  - Every semester, at least one student makes this mistake. You need to write the steps first, THEN do the walkthrough. May be helpful to do at different times.
- The walkthrough does not test real users on the system. With a walkthrough you can potentially evaluate the interface by imagining the behavior of entire classes of users, not use one unique user.

Goal: Create UML diagram in DIA

- Want to create a simple UML diagram:



Representative Task

What Persona?

- Student taking software engineering
- Basic familiarity with various kinds of software, including drawing programs
- Understands the parts of a UML diagram (i.e., the tool is not teaching UML, but helping to easily create diagrams)

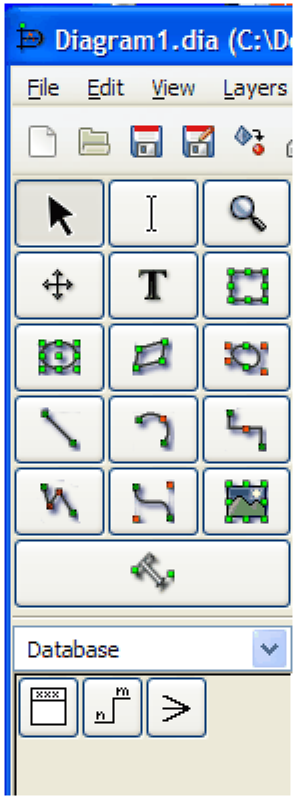
- As you read the example, think about how the analysis would change if these user assumptions were not true.

Steps

1. Put in UML mode
2. Add parent class (Student)
   A. Select class tool
   B. Draw class onto canvas by clicking
   C. Change class name
3. Add name as private String
   A. Bring up dialog, click on Attribute tab
   B. Click New
   C. Enter name
   D. Change visibility to Private
   E. Click OK
4. Add public method addCourse (String parameter)
   A. Click on Operations tab
   B. Press New
   C. Enter method name
   D. Click New parameter
   E. Enter parameter name (course)
   F. Enter parameter type (String)
5. Add CSMajor and MathMajor as children
   A. Create CSMajor and MathMajor classes, as above
   B. Line them up on the canvas
   C. Select Generalization tool
   D. Drag mouse from parent class to one child
   E. Use Zigzagline to connect to second child

Step 1: UML mode

- Screen comes up in database mode



I'm thinking: I want to create a UML diagram

Action:

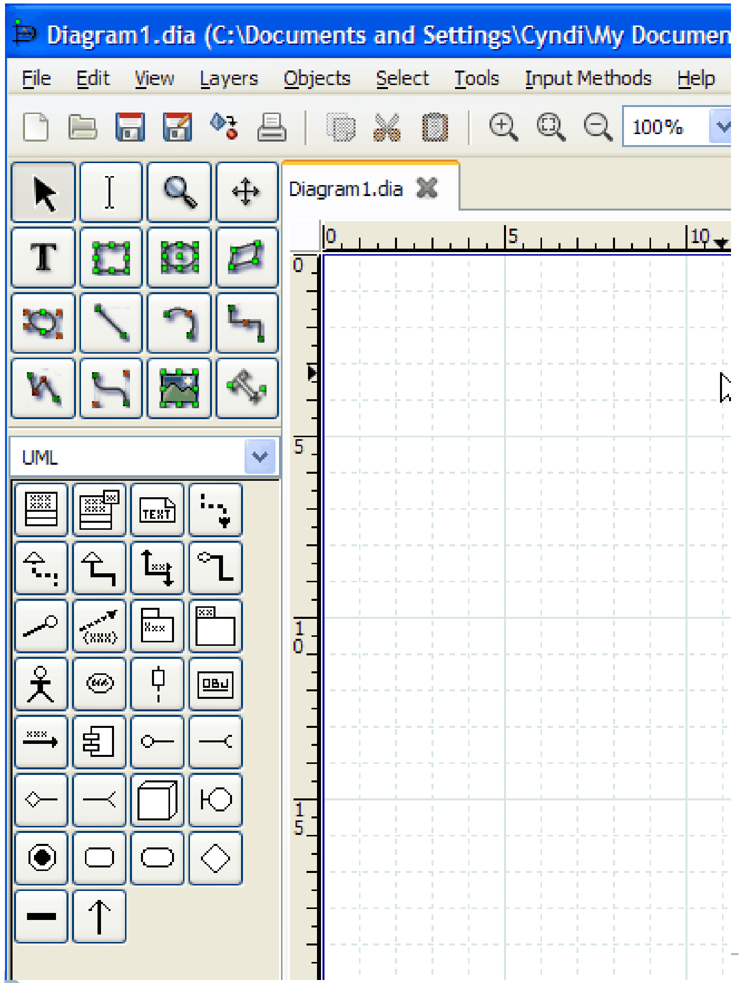
- I see a lot of symbols that aren't UML.
- I look through the menus, don't see UML.
- Finally notice drop down with Database. I try it. Now I see UML.

Recommendation:

- Highlight the drop down. It's not hidden, but its location in the middle of the screen makes it much less obvious.
- Also, add a Diagram Type option to one of the menus, maybe Select.

Step 2: Add parent class (Student)
Step 2A: Select class tool

- Now the UML menu is available.



I'm thinking: I want to draw a box for Student. Student is the parent class.

Action:

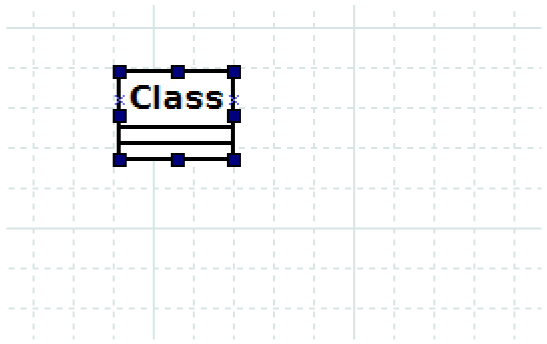
- Top symbols all seem generic, I don't pay much attention.
- The symbols under UML are more relevant.
- First one looks like the right kind of box (or maybe second one).
- I move mouse over first symbol. Tool tip for first one says class. I click on it.

Recommend:

- Tool tips are effective.
- Class is first icon, good because most UML diagrams probably begin with Class.
- No issues with this step… but it *might* be good to highlight the Class button, if the drawing is empty.

Step 2: Add parent class (Student)
Step 2B: Draw class onto canvas by clicking

- Now I've selected the class tool



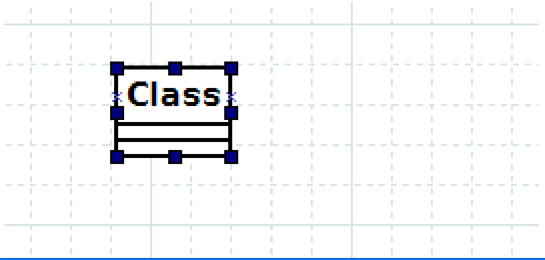
I'm thinking: OK, now I want to add Student to my diagram. Do I click or drag?

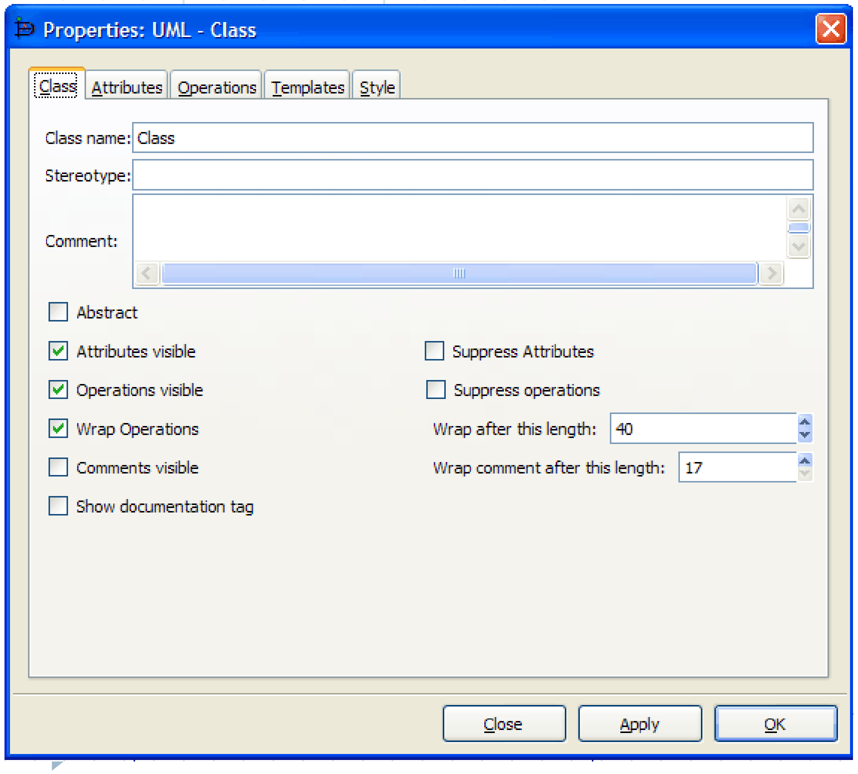
Action:
- I click on the canvas.
- Class is added, with name Class.

Recommend:
- This seems clear (for users with some drawing experience), no recommendation

Step 2: Add parent class (Student)
Step 2C: Change class name

- Now I've added the class to my drawing





Class name: Class
Stereotype:
Comment:
☐ Abstract
☑ Attributes visible
☐ Suppress Attributes
☑ Operations visible
☐ Suppress operations
☑ Wrap Operations
Wrap after this length: 40
☐ Comments visible
Wrap comment after this length: 17
☐ Show documentation tag

I'm thinking: I want to change the name to Student.

Action:

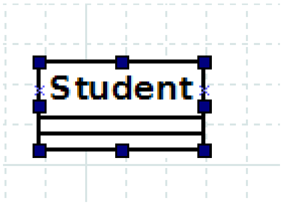
- I click on name (thinking I might edit in place). No action.
- I look at Edit menu, but nothing obvious.
- I double-click where it says Class.
- Dialog comes up.
- First text field is Class name. I enter Student. Press OK.
- Class name is changed to Student.

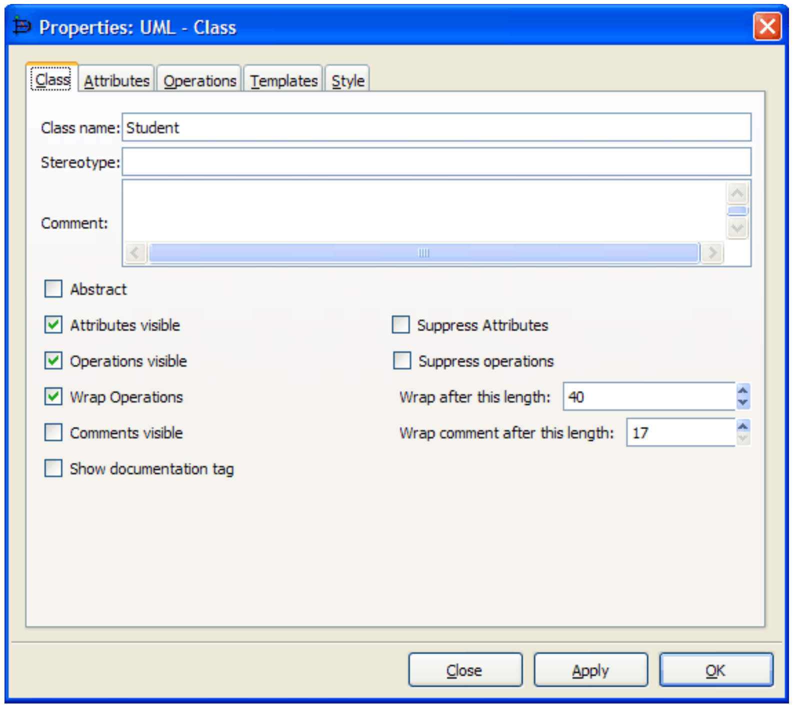
Recommend:

- It might be nice to add a Properties option to the Edit menu, as some users are very menu oriented.
- A more advanced program might allow the user to click and edit things like names… but DIA is a free program, and it doesn't take long to figure out the double click. So OK.

Step 3: Add name as private String
Steps 3A: Bring up dialog, click Attribute tab

- Now I've changed the class name to Student





Properties: UML - Class

Class | Attributes | Operations | Templates | Style

Class name: Student

Stereotype:

Comment:

☐ Abstract
☑ Attributes visible
☑ Operations visible
☑ Wrap Operations
☐ Comments visible
☐ Show documentation tag
☐ Suppress Attributes
☐ Suppress operations
Wrap after this length: 40
Wrap comment after this length: 17

Close | Apply | OK

I'm thinking: OK, I want to add my fields. Editing in-place is not an option, but there were a lot of options on that dialog I just used.

Action:

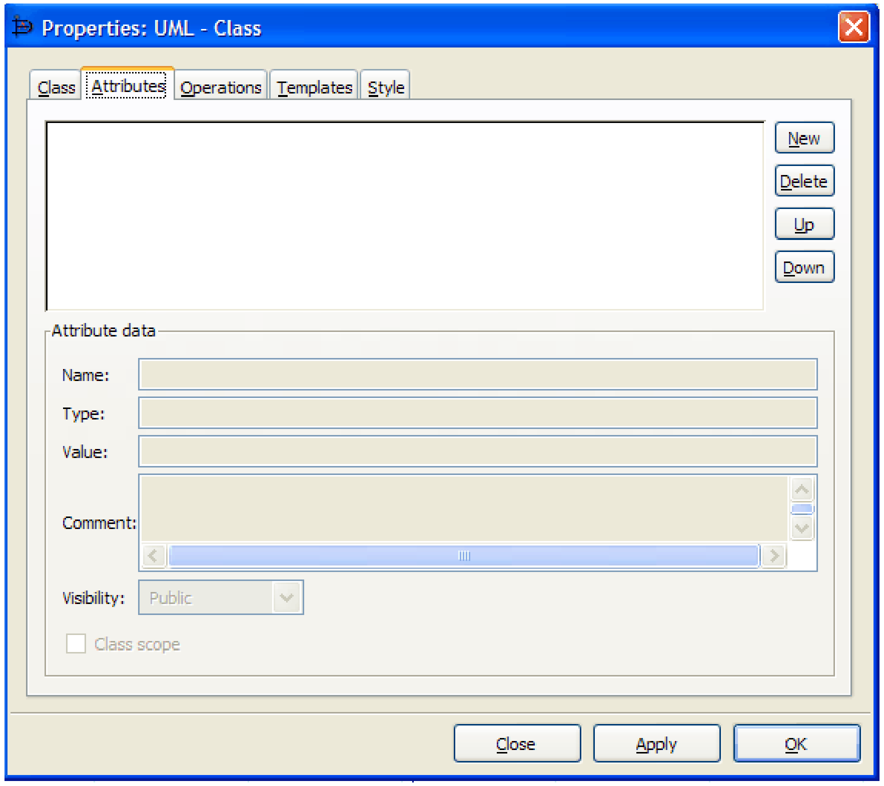
- I double-click on Student class.
- Dialog appears.
- Checkboxes don't seem to apply. I notice Attributes (which I recognize as synonym for fields – this could be a vocabulary issue for some users). Click on Attributes tab.

Recommend:

- Tabs probably OK for experienced users. Would a novice notice? Different styling might make tabs more obvious.
- Dialog has options I don't understand (e.g., Attributes visible vs. Suppress Attributes, Wrap options). Visual icons might help explain these options.
- It would also be nice to bring up the dialog in Attributes mode if I click on the Attributes part of the drawing.

Step 3: Add name as private String
Steps 3B:Click New

- Now I'm at the correct dialog



I'm thinking: I want to type in the variable info.

Action:

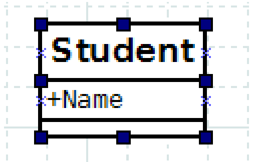
- I try to type in Name: field, but it's grayed out.
- I consider just typing into the big text box, but that doesn't seem right.
- I notice New, figure that's what I need.
- Click it, I'm able to enter a Name and Press OK.

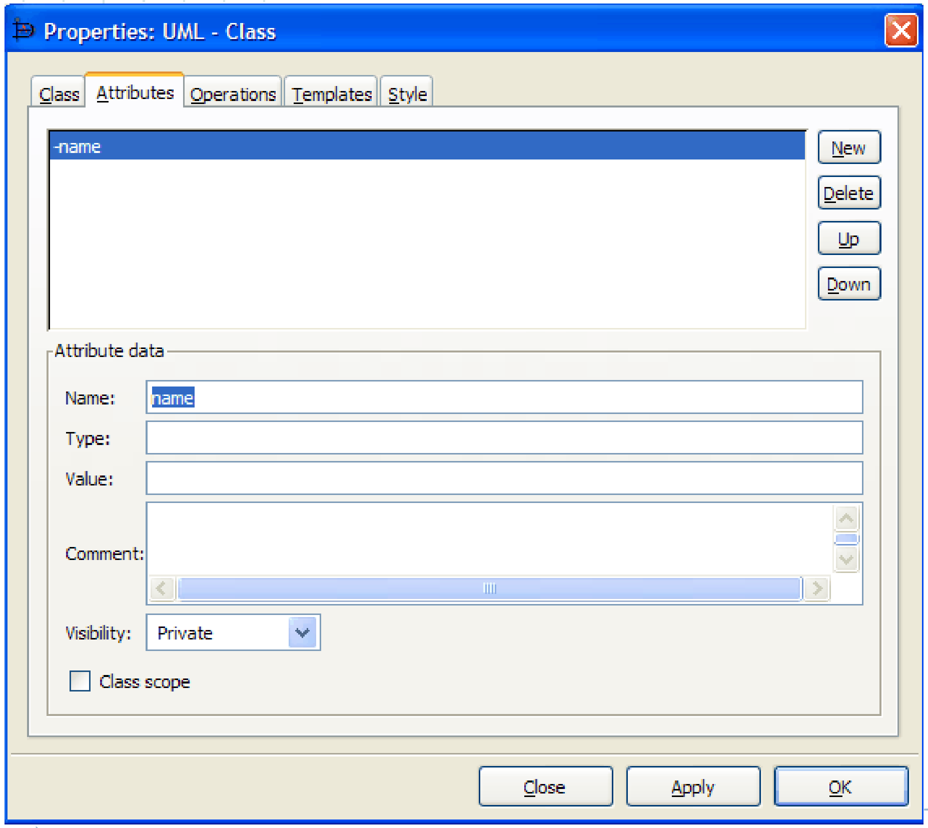
Recommend:

- We read left-to-right. I would probably put buttons on left side of text area. Maybe put default text such as "No attributes defined" or "Click New to add attribute" in the text area.
- If no attributes defined, gray out Delete, Up and Down buttons. Would make New stand out.
- What's the difference between Apply and OK? Confusing, but I've met my goal so I ignore for now.

Step 3: Add name as private String
Steps 3B:Change visibility to Private

- Now my attribute is listed, but it has a +





-name

Attribute data

Name: name

Type:

Value:

Comment:

Visibility: Private

[ ] Class scope

I'm thinking: I missed something.

Action:

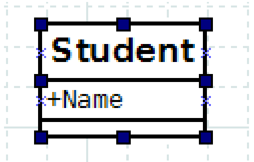
- I bring dialog back up
- Click on name
- I quickly notice Visibility, change to Private

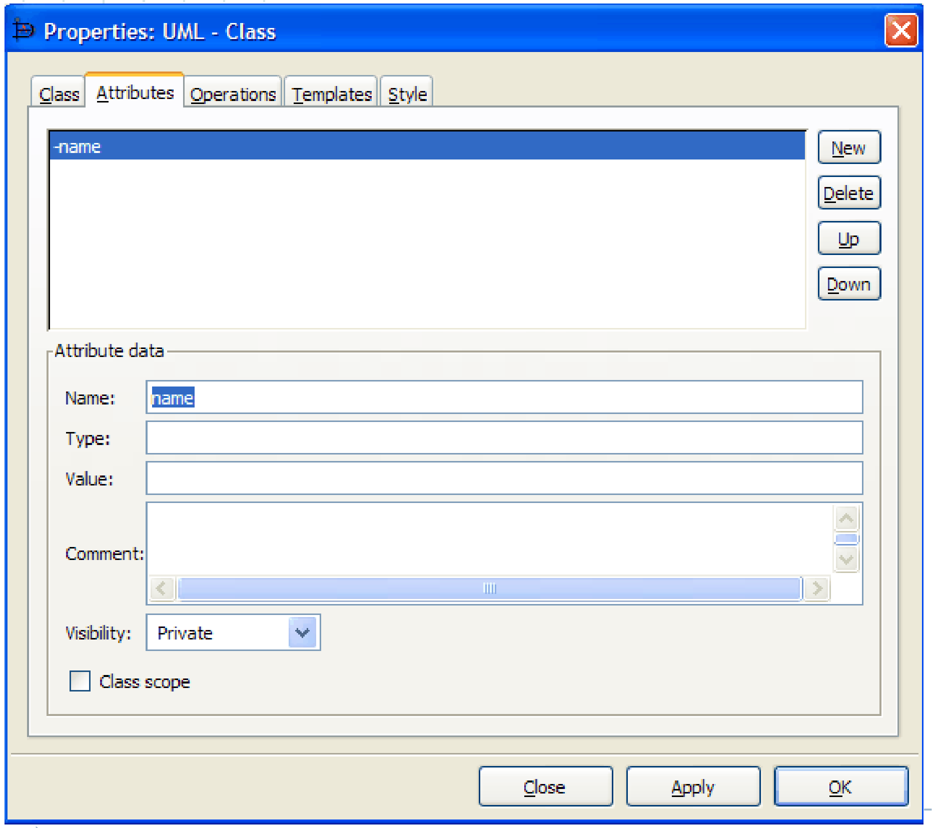
Recommend:

- I would default to Private (that's normally recommended except for constants)
- I would move Visibility higher in list, after Type or Value

Step 4: Add name as private String
Steps 3B:Change visibility to Private

- Now my attribute is listed, but it has a +





-name

Attribute data

Name: name

Type:

Value:

Comment:

Visibility: Private

☐ Class scope

I'm thinking: I missed something.

Action:

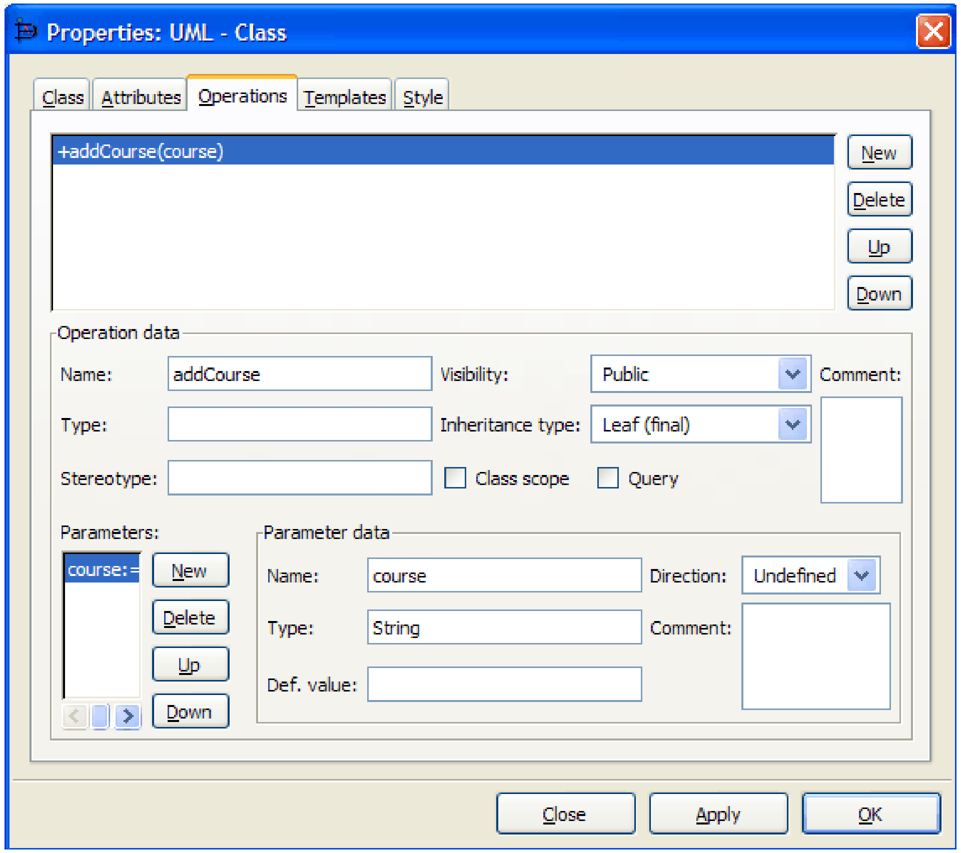
- I bring dialog back up
- Click on name
- I quickly notice Visibility, change to Private

Recommend:

- I would default to Private (that's normally recommended except for constants)
- I would move Visibility higher in list, after Type or Value

Step 4: Add public method addCourse Steps 4A – 4E

- Now my attribute is listed, but it has a +



+addCourse(course)

Operation data

Name:	addCourse	Visibility:	Public	Comment:
Type:		Inheritance type:	Leaf (final)	
Stereotype:		☐ Class scope	☐ Query	

Parameters: course:=

Parameter data

Name:	course	Direction:	Undefined
Type:	String	Comment:	
Def. value:			

I want to add a method.

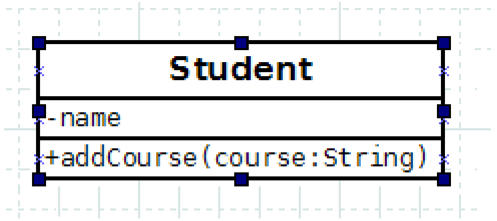
Action: I know now to look at the tabs. Methods is not there, but Operations is. Screen operation is similar to Attributes, so I immediately press New. I then enter the method Name. I press New under parameters. I enter the Name and Type.

Recommend:

- Similar suggestions as for Attributes.
- I'm not sure what some options are (e.g., Query) but I ignore for now.

Step 5: Add CSMajor and MathMajor as children Step 5A:

- Now I have a fully defined parent class



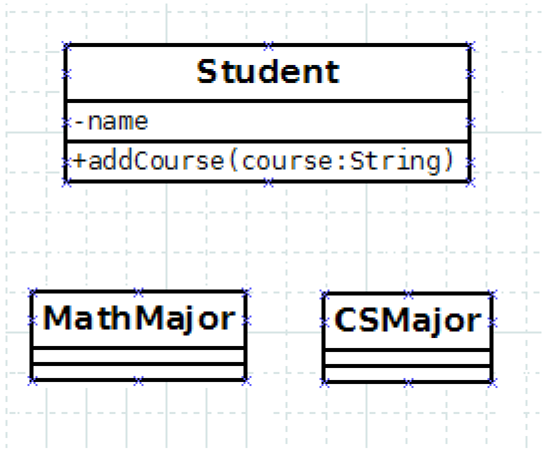
I'm thinking: I know how to create classes, first I need to create the two children.

Recommend:

- No recommendation

Step 5: Add CSMajor and MathMajor as children Step 5B: Line them up on the canvas

- Now I have 3 classes



I'm thinking: it was easy to add the classes, but now I have to put in the relationships. First I want them lined up below the parent.

Action:

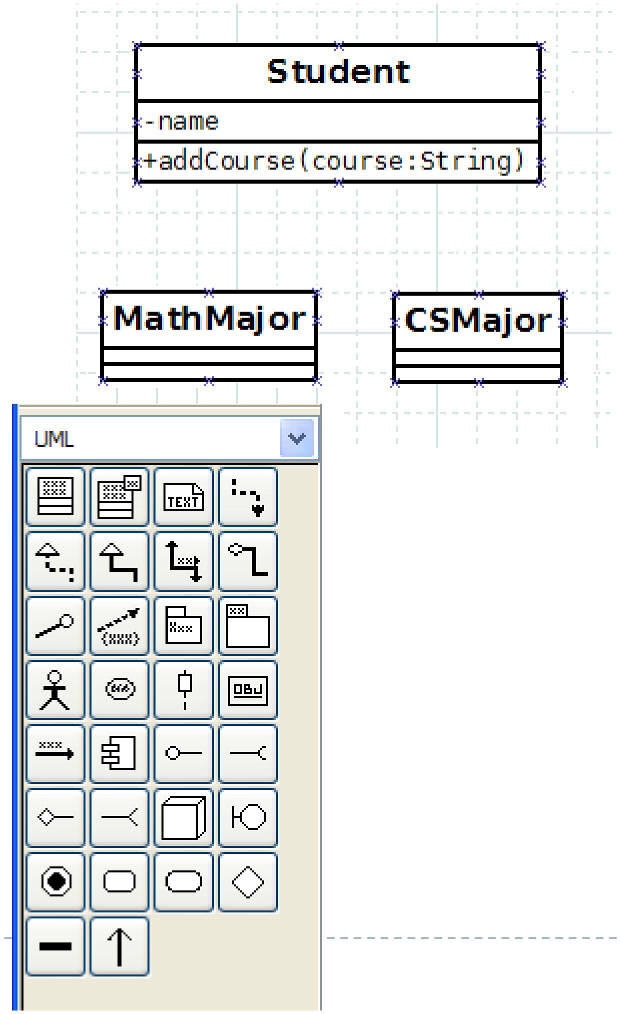
- The canvas is like most drawing programs, so I just click on the objects and move them.

Recommend:

- No recommendation

Step 5: Add CSMajor and MathMajor as children
Step 5C: Select Generalization tool

- Now I have 3 classes lined up



I'm thinking: I need to add an inheritance relationship from the parent to each child.

Action:

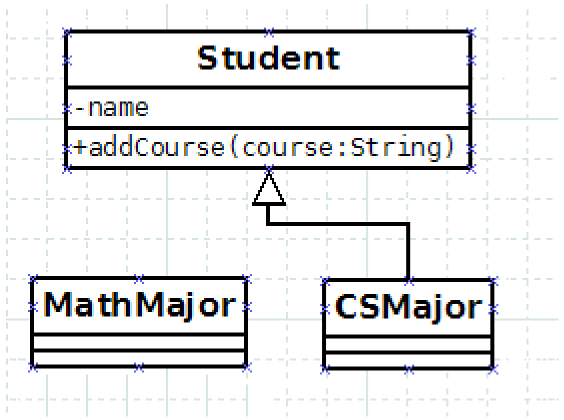
- Notice that the UML toolbar has a tool in the 2nd row that looks like generalization. Tool tip confirms.

Recommend:

- No recommendation

Step 5: Add CSMajor and MathMajor as children
Step 5D: Drag mouse from parent class to one child

- Now I have 3 classes and have selected inheritance tool



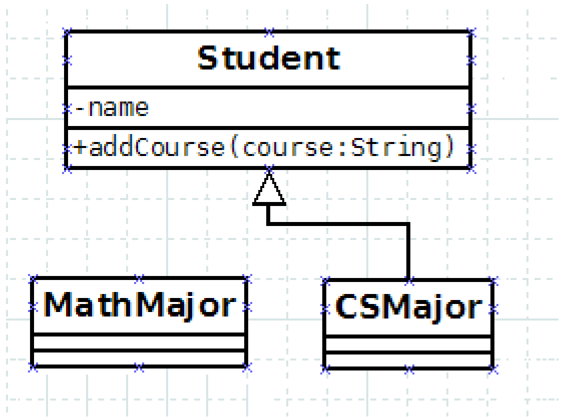
I'm thinking: This looks like a typical drawing tool. I should draw from the parent to the child (the child "looks up" to the parent… and the icon reinforces this)

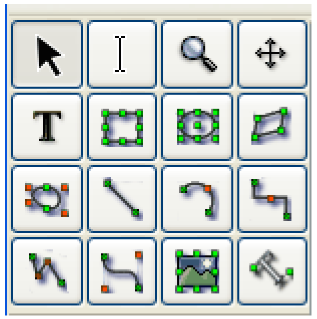
Action:
- Use tool to draw as expected. As I'm drawing I notice the connection points on the sides of the classes. Line snaps into place.

Recommend:
- No recommendation

Step 5: Add CSMajor and MathMajor as children Step 5E: Use Zigzagline to connect to second child

- Now I have 3 classes and one inheritance relationship





I'm thinking: There should be an easy way to connect a second child.

Action:

- I try to click on existing line, but don't see any way to extend it to the 2nd class. I look at other tools at top of program. I notice the jagged line (tool tip says Zigzagline). Click on that, use to update drawing. *

Recommend:

- The drawing looks OK, but there doesn't seem to be any semantic meaning. It would be great to click on triangle, click on 2nd child, have the tool generate the line.

- * There may be a better way to do this, but I haven't found it.

Task Complete!

- And we have a number of potential recommendations.