A Brief Introduction to Visual Paradigm (VP) for UML

1. Introduction

Visual Paradigm for UML is a CASE (Computer-Aided Software Engineering) tool that supports numerous modelling standards, including most notably:

- UML – Unified Modelling Language
- SysML - Systems Modelling Language
- ERD – Entity Relationship Diagrams

Visual Paradigm is not just a drawing tool but also a sophisticated modelling tool which facilitates the creation of system, software engineering and business models. You are encouraged to use the online resources and the Visual Paradigm Online tutorials (available here: https://online.visual-paradigm.com/diagrams/tutorials/) to familiarise yourself with these features in your own time. This university has the academic edition on their machines, however, for a combination of licensing and compatibility reasons it is best for us to use the online edition, available here: https://online.visual-paradigm.com/w/qtbkbdml/drive/#infoart:proj=0&dashboard

Please register and make an account with your university email address. Click on the user account icon at top right and select "Sign up". Once done, go to the "Diagram" tab and select **Class Diagram** (not ERD!!).

In this unit we will concentrate on three diagrammatic models:

- Entity Relationship Diagrams – model data/information within a business context. Usually used for creating a design suitable for implementation in a relational database.

You may cover further UML diagrams in the second year but feel free to explore them now.

Entity Relationship Modelling with Visual Paradigm Online

2. Introduction

This guide is intended to introduce you to the process of creating Entity Relationship Diagrams using Visual Paradigm Online in Unified Modelling Language (UML). If you have not used Visual Paradigm before it is recommended that you first complete the "Introduction to Visual Paradigm Online & Use Case Diagrams (UCDs)" worksheet available on Moodle.

Entity Relationship Modelling is a modelling technique used to describe the data or information elements of a business system. The main output of the modelling technique is an Entity Relationship Diagram or ERD. Entity Relationship Modelling is widely used to create a model of an information system that can be used as a design specification for a relational database. Therefore, ERDs are commonly associated with database design.

3. An Important Note on Notation

Although the core concepts of ERDs have remained relatively constant, over the years there have been several different ERD notations (i.e. sets of symbols, diagram layouts and rules). You are likely to encounter more than one notation when working in the industry depending on the age of the modelling material in the organisation. There has been a general move towards UML in the software engineering industry and, although designed for object-oriented software engineering, UML Class Diagrams are conceptually equivalent to ERDs in that they show abstract 'things' (i.e. classes/entities) which are identified in the ERM process.

We will therefore use Visual Paradigm's UML Class Diagram tool to create our ERD but will only use a selection of the relevant Class Diagram elements. For reference, a mapping of terms between ERD and UML Class Diagram notation is included below in table 1.

Entity Relationship Diagrams	UML Class Diagrams
Entity	Class
Relation	Association
Cardinality	Multiplicity
Optionality	

Table 1: ERD <> Class Diagram notation mapping

Note that Visual Paradigm does have a separate Entity Relationship Diagram tool which uses a notation known as 'crow's foot' or Barker's notation. You may wish to try this tool in your own time to familiarise yourself with the notation but bear in mind that this notation may be considered outdated in the industry, so we use the class diagram-like notation.

4. Creating an Entity Relationship Diagram

TASK 1: Go to: https://online.visual-paradigm.com/drive/#diagramlist:proj=0&new **. Open up a blank Class Diagram. From the webpage, select 'Class Diagram' from the scroll list on the left hand side. Several options will appear in the main window showing different examples. Select the top left option shown in Figure 1 below the text that says "Build from scratch".**

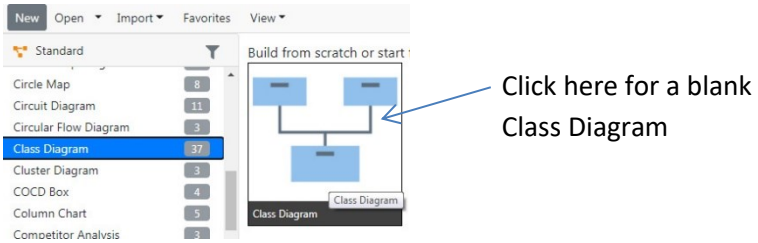


Figure 1: Select Class Diagram

As shown in Figure 2, a blank canvas will appear with a Class Diagram specific toolbox just to the left. The main diagram elements that we will use are Classes (to represent Entities) and Associations (to represent Relations: one to one, one to many, many to many). Most of the other diagram elements are relevant only to Class Diagrams, not ERDs.

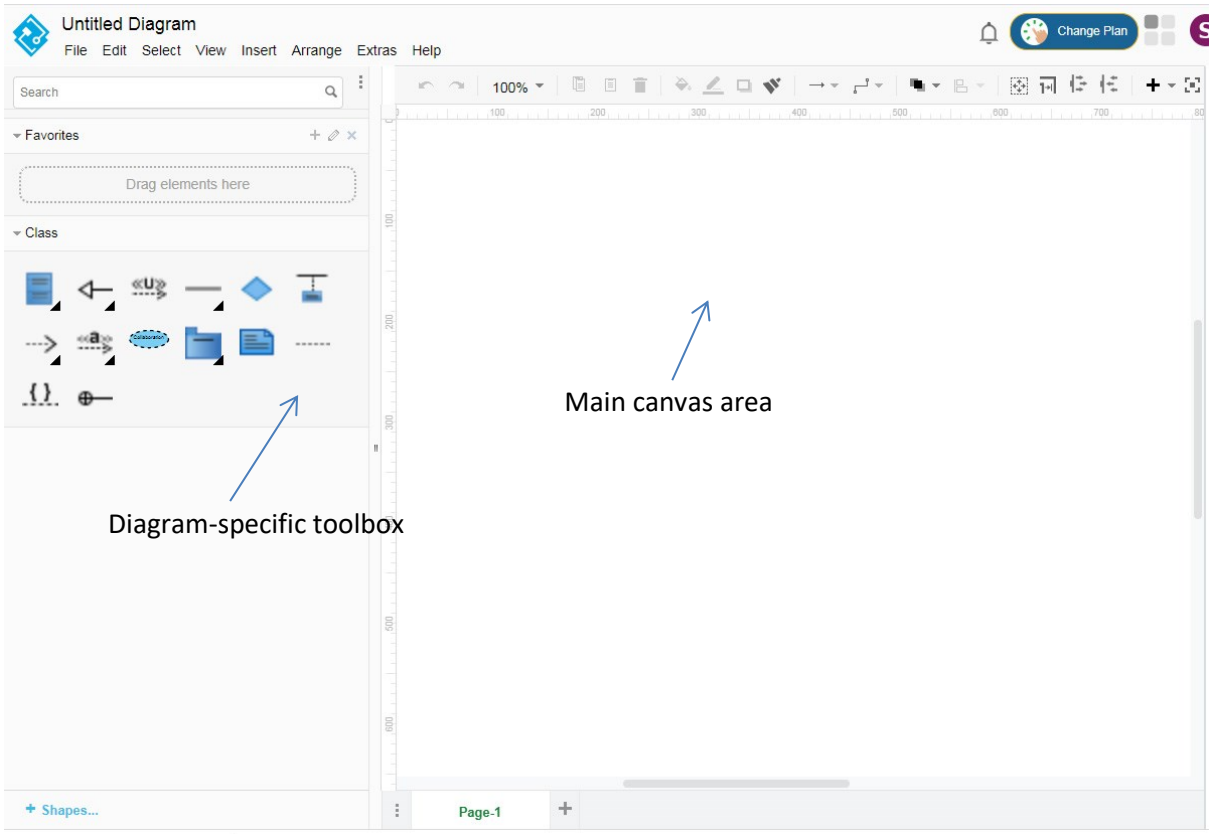


Figure 2: Blank Entity Relationship Diagram (Class Diagram) canvas

5. Adding Entities to the Diagram

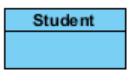
TASK 2: Add entities to the ERD

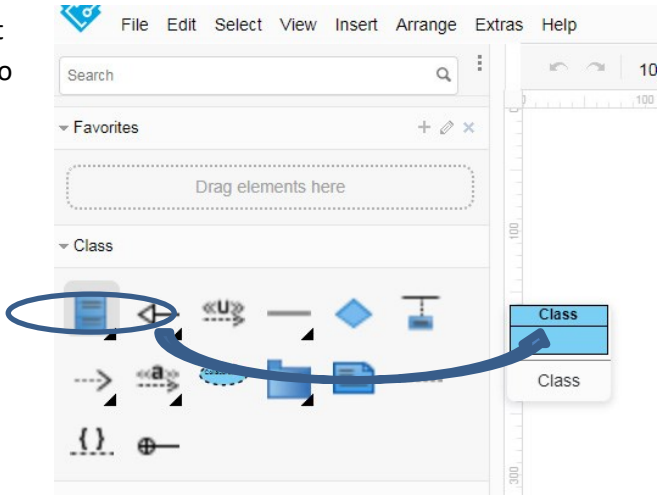
First we'll add an entity named 'Student' to our diagram.

Click the Class icon from the toolbar on the left and then click on the canvas where you want to place the new entity.

Once placed on the canvas you can type the name of the new entity. In this case, name it 'Student' and the press enter.

Once created the entity should be shown like this:





Repeat this process to add another entity called 'Tutor'.

Figure 3: Creating an Entity named Student

6. Adding Relationships between Entities

TASK 3: Add relationships to the ERD

Figure 4: Student & Tutor Entities

You should now have a diagram with two entities as shown in Figure 4. We need to represent the relationship between students and their personal tutors. We will assume that this is a 'one to many' relationship i.e. a student has a single personal tutor and a tutor can supervise many students. We will also assume that a student must have a personal tutor but that some tutors may not supervise any students. Therefore, another way to describe this relationship is that a student has a minimum of 1 tutor and a maximum of 1 tutor, and that a tutor has a minimum of 0 students and a maximum of many students.

We create relationships by using the Visual Paradigm 'Association' diagram element. As with other diagrams this can be achieved by clicking on the

button on the toolbar (it will not have the word next to it). Then click and hold on one entity, drag to the other entity and let go (of the click). This will create the line between the two entities. Note that if your entity is full of attributes and you drag onto an attribute the association will not be established. To deal with this click and drag from entity name to entity name.

You can then name the relationship, if you wish. Double click in the middle of the association line and type the label 'Supervises' to indicate that the tutor supervises students (see Figure 5).

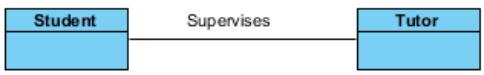
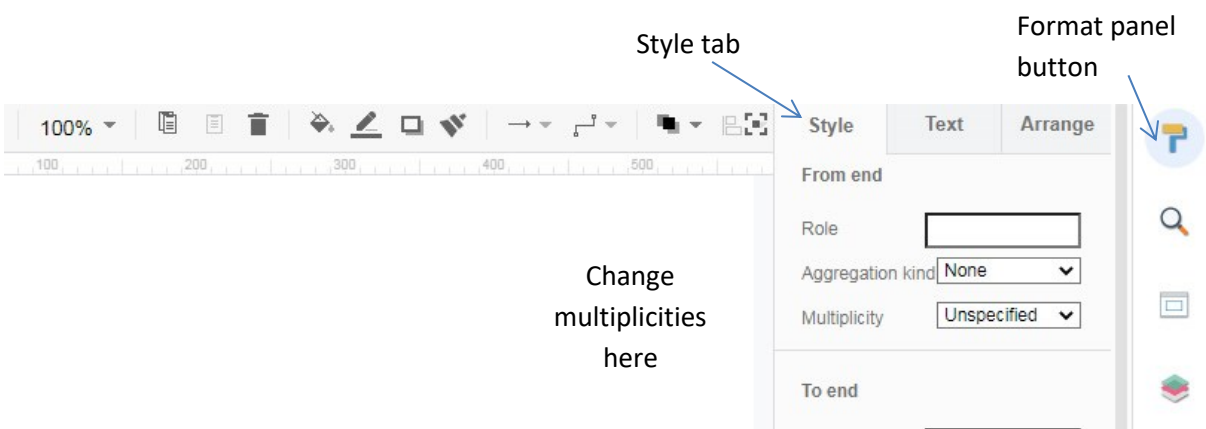


Figure 5: 'Supervises' Relationship

TASK 4: Further specify relationships on the ERD

Next, we need to add details about the numbers involved in the relationship. This is referred to as cardinality and optionality in Entity Relationship Modelling and multiplicity in UML.

Select the association and then click on the format panel button and select the "Style" tab (see Figure 6). (The style tab is context aware and so it will show different things depending upon what is selected.)



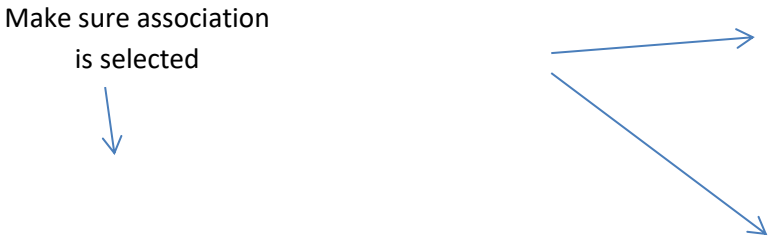


Figure 6: Adding Multiplicities

Change the Multiplicity to 0..* at the student (From) end of the association, and to 1 at the tutor (To) end. In other words, a student must have 1 and only 1 tutor, and a tutor can supervise a minimum of 0 and maximum of many students. You should have a diagram like the one shown in Figure 7.

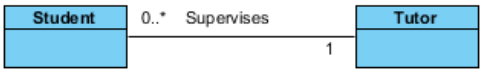


Figure 7: Entities with Relationship

Although this is a very simple system which we have modelled, the technique is the same for all other entities and relationships that you may wish to add.

During the initial stage of entity relationship modelling it is common to create a conceptual model in this manner or, more likely, several iterations of the design in order to identify all possible entities and relationships and to get a general understanding of the system design. However, once the entities and relationships have been defined it is then useful to add more detail.

7. Adding Attributes to the Entities

TASK 5: Adding attributes to the entities

We will now add some attributes to the Student entity. This will allow us to design what data will be kept about each student and to specify primary and foreign keys. We can also optionally specify which datatypes each of the attributes will use.

Right-click on the Student entity, and select 'Add Attribute'. This will add a new attribute to the Student entity. Type 'studentNumber' (note attributes should start with a lower case letter, contain no spaces and that each subsequent word should be capitalised) then press Enter. Resize the entity box using the mouse to accommodate the attribute name.

Note that when resizing the entity, it should look as shown in Figure 8. If it does not, move your mouse over the name of the entity and click the mouse once.

If you don't do that you might end up with something like that shown in Figure 9. If you do end up with that, just push the escape key and try again.

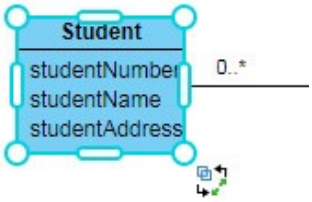


Figure 8: Entities with Relationship

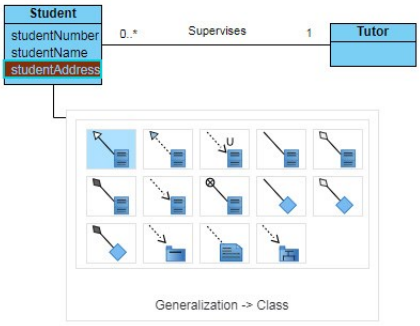


Figure 9 (left): This different options menu is not what you want. It represents another way of making associations that we are not covering in this unit.

Follow the same procedure to add these attributes:

studentName
studentAddress
studentPhoneNumber
studentTutor

If you need to edit any of the attributes just double click on them. We now need to set the studentNumber attribute as a primary key and studentTutor as foreign key. To do this we will use UML Stereotypes. To do this in VP Online you just type it in by editing the attributes. We do it with following text "<<PK>>" for a primary key. You should end up with a diagram like that shown in Figure 10.

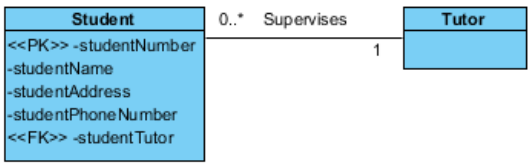


Figure 10: Diagram with Student Attributes

8. Finishing Off

You can now finish off the diagram yourself. Add appropriate attributes to the Tutor entity and define an appropriate primary key. Remember that the primary key of the Tutor entity will relate to the 'studentTutor' foreign key of the Student entity. Also feel free to add additional entities and relationships for additional practice.

Be sure to save your work when you finish (see below).

9. Getting an Account

To save you need to get an account. (To avoid problems with backwards compatibility we do not advise students to use the version of Visual Paradigm installed on university computers, but rather to use the online version that is freely available.)

To get an account on Visual Paradigm you will need to register using your student email address (or personal details, if you wish). Sign up here: https://online.visual-paradigm.com/subscribe.jsp . Once signed up you can login and this allows you to save to your device.

10. File Handling

Once you are signed up you can save your diagram to your device. Select the ☰ icon and then select: File > Save as… > Device (See Fig 11). You should then be able to save to your "One Drive" space. To open a file, select the ☰ icon and then select: File > Open.

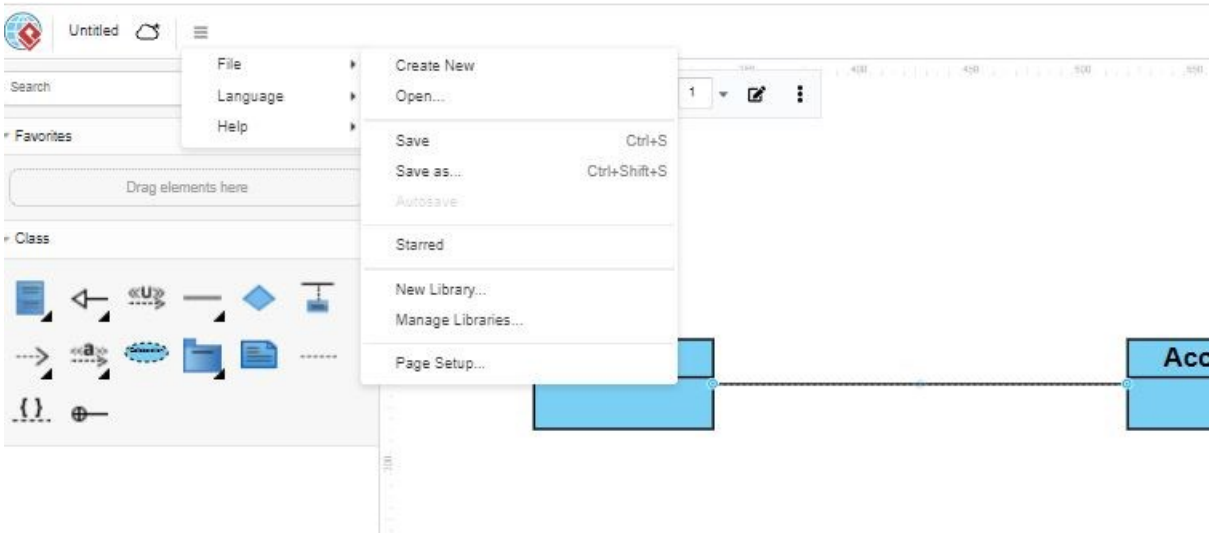


Figure 11: Saving a diagram file